

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 10, 2011

S. Brim  
M. Linsner  
B. McLaughlin  
K. Wierenga  
Cisco  
October 7, 2010

Mobility and Privacy  
draft-brim-mobility-and-privacy-00

Abstract

Choices in Internet mobility architecture may have profound effects on privacy. This draft revisits this issue, stresses its increasing importance, and makes recommendations.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 10, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. The Risks of Being Traceable . . . . .	3
3. Basic Mobility Requirements . . . . .	4
4. Avoiding Making a Mobile Node Traceable . . . . .	5
5. Recommendations . . . . .	7
6. IANA Considerations . . . . .	8
7. Security Considerations . . . . .	8
8. Acknowledgements . . . . .	8
9. Normative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

Significant steps are being taken right now to make the Internet's architecture more scalable and robust in routing, addressing, multihoming, mobility, and locator/identifier separation. However, since the Internet is rapidly becoming an essential part of daily life for people around the world, our architectural changes need to take fundamental social issues into account as a primary consideration. One of those is privacy, and in this case particularly privacy of geographic location. If we do not, we run the risk of colliding with established IETF principles (see for example [RFC3693]) as well as public policy in many countries around the world.

When the Internet was designed, IP addresses were associated with timesharing machines and not with particular users. In the 1980s it began to be likely that a device and thus an IP address would be associated with a single user. Now a single IP address is very likely to be associated with a single human being. Meanwhile, at the top of the stack, there has been a convergence of life functions using single devices using single addresses. A person now uses his or her personal device and associated IP address for many activities: work, shopping, talking, exchanging mail and files, reading, listening to music, etc.

It is this convergence at both the top and bottom of the stack -- to a single person per device and to many applications on that device -- that makes the social issues more and more significant in IETF work. People use the Internet for many, more personal, activities than before. The Internet needs to fulfill the obligations expected of a communications system essential to modern human society. Our lower layer protocol designs have privacy implications beyond their intended scope.

## 2. The Risks of Being Traceable

Issues with revealing geographic location are well-established elsewhere. For example the RAND review of the European Data Directive [RAND-EDPD] points out that "the interpretation of location data (e.g. which locations are visited, suggesting which shops are frequented, and which products and services are bought), may in the future permit the identification of the health, social, sexual or religious characteristics of the data subject" (section 3.3.1). The less well-known problem that this document focuses on is tracing the movement of mobile devices. Because mobile devices are used for so many things, any possibility of tracing them has significant, probably unpredictable, social implications, perhaps more so than

revealing a single location. If an association can be made between a mobile device and a person at any location, if that device can be traced to a different geographic location then the association with the person can be inferred, usually correctly, even if the person believes they are anonymous at the new location. Consider scenarios such as:

- o You are looking for a job, interviewing at other companies over your lunch hour, but you don't want your current management to know.
- o You are planning a surprise gift or party for your spouse and are visiting specialty stores.
- o You are a journalist gathering information on a corrupt politician from sources who wish to hide that they are dealing with you.
- o You are infiltrating an organized crime ring and don't want them to know when you sneak in the back door of police headquarters.
- o You are a very famous person trying to avoid paparazzi and assassins who are able to find you sporadically.

Mobility mechanisms need to take this issue into account. Obviously a mobile node must be reachable somehow, but a mobile node must be able to hide its actual movement from public view if it wishes.

### 3. Basic Mobility Requirements

A mobile node may need to be reachable by others, or it may act purely as a client of Internet-based services. Even if it is purely a client, it still needs at least two things:

- o An authentication and authorization identifier that it can use with each access network it connects to. (Not required for open access networks.)
- o A Layer 3 way for its correspondents to get packets back to it. This may no longer be simple due to potential innovations in routing architecture.

In addition, if the mobile node wants to be reachable as a peer or to offer services, it needs a few more things:

- o An identifier (or identifiers) by which the node may be found by others, and a mechanism by which this identifier can be mapped to IP addresses/locators. Examples are domain names, SIP URIs, and

the corresponding services.

- o An IP address/locator for initially contacting the mobile node. This does not have to be associated with the mobile node's actual topological location. It can instead be associated with a rendezvous point or agent.
- o A mechanism for "route optimization", whereby such an agent can be eliminated from a data path between the mobile node and a correspondent.
- o An identifier or identifiers by which the mobile node can authenticate itself to its correspondents during initial contact, route optimization, and/or change of topological location. These identifiers can be at any layer, from 2 to 7. They can be associated with the mobile device's whole IP stack, individual transport sessions, or individual application instances.
- o Identifiers by which the mobile node can be referred to by third parties.

If all mobile nodes are reduced to being clients only -- if they are willing to register with servers in order to use the Internet and have others be able to reach them -- then there are fewer requirements. However, over the evolution of the Internet we have seen several times that it is not good to give up the symmetry of Internet communication and "permission-free" networking, i.e. the ability for anyone anywhere to communicate as a peer with other nodes on the Internet. For the rest of this document we assume that the IETF still wants to retain this model.

Every identifier listed above has a scope in which it needs to be known, but it is only required to be known in that scope. For example, an access authentication identifier only needs to be known to the mobile node, the access network, and a trusted third party (a mobile node's home network administration, or a bank, etc.). A session identifier only needs to be known among the parties using it, but not by the access network.

#### 4. Avoiding Making a Mobile Node Traceable

As a mobile node moves, if L3 or higher layer mobility mechanisms are used it will change its IP addresses/locators. The Internet already has sophisticated publicly available services for determining where a node is based on IP address alone. These mechanisms are not always precise or accurate, but they are in very many cases and even imprecise information is information. Protocol designers must assume

that whatever IP address or locator a node has, it is likely that there is a service to turn that into a geographic location.

The tracing problem occurs when it is possible for a third party to correlate IP addresses/locators and something unique about the mobile node. Data can be gathered either through monitoring traffic or by accessing public information. It does not have to be done continuously -- periodic snapshots can make the mobile node just as vulnerable. Once the data is gathered, the third party can search for correlations.

Using identifiers for multiple purposes makes leakage of tracing information more likely. Different entities in different scopes may know different things about a mobile node or a person. Using overlapping identifiers mixes scopes and may make new, perhaps unexpected, correlations easier. For example if an access identifier such as a mobile phone's IMEI (hard-coded and not changeable, primarily used for access authentication) is also used for session continuity, or is registered in an Internet database service that is publicly accessible, changes in that device's IP addresses (and thus geographic location) can be traced.

Long-lasting identifiers make correlation easier as a device moves. They should not be used in scopes where they are not necessary.

The biggest concern is if information that makes a mobile node traceable is required to be publicly available in order for the Internet to function. If it is, it can be accessed not only without the mobile node's consent but even without its knowledge, perhaps without any audit trail of who is accessing the information that could be looked at after the fact. Some architectures for mobility and/or routing and addressing described in [I-D.irtf-rrg-recommendation] assume the use of DNS or other public mapping systems. In these, the mobile node is required to publish a mapping between its identifier and its current IP addresses/locators in order to be reachable, even if a mobile node is acting purely as a client (because otherwise packets would not get back to it). This architectural assumption removes all of the mobile node's freedom of choice about how much confidentiality to preserve -- either it exposes all of its movement to all of the world or it is simply not reachable. Public information systems like DNS are not designed to support confidentiality.

MIPv6's "home agent" [I-D.ietf-mext-rfc3775bis] is an example of how to avoid this problem: Contact with a mobile node is initially through a home agent, a rendezvous point for both data and control traffic. The home agent acts on behalf of the mobile node and encapsulates traffic to it. After an exchange of packets, the mobile

node may decide, on its own, if it wants to reveal its topological location, and thus probably its geographic location, to the correspondent node. It controls its own location information. The decision to reveal it can be based on anything, including local policy.

The principle of hiding information that can expose geographic location in both data and control planes, and deferring revealing more until the mobile node or its agent decides what it wants to do, is essential. This can be included in any mobility architecture that is designed to allow it and does not insist on exposing location to a wide audience in order to gain efficiency. The obvious way to do it is an indirection mechanism such as a home agent, but this is just one way to do it. Any way will do.

Monitoring is a more subtle issue than exposure in public services, but still real, even if the mobile node is client-only. If packets contain an identifier that uniquely identifies the mobile node for some period of time, someone able to gather data on packet traffic can easily trace the mobile node's movements as the IP address/locator changes. It is not necessary for the watcher to be able to gather this information in real time if it can access logs gathered by others. Here, approaches to the problem are more difficult to define because there is a conflict between three goals: to avoid overhead, to preserve session continuity with low delay, and to keep control over location information. Some designs such already try to find their balance. All protocol work should consider the tradeoffs with privacy and explicitly find a balance point.

## 5. Recommendations

Members of the Internet community who are creating or reviewing proposed architectural changes, particularly in mobility but also in other areas that impinge on mobility such as routing and addressing, should consider the following points:

- o Architectural changes MUST avoid requiring exposing a mapping between any of a node's identifiers and IP addresses/locators to unknown observers. If they require exposure, they will experience a head-on collision with basic principles of the IETF and with privacy policies around the world. It will simply not be acceptable to require the loss of this much individual privacy.
- o An architectural proposal MAY make it possible to use public information systems to optimize traffic flow, but ideally it should do so without sacrificing privacy. If it cannot do so without sacrificing privacy, the default case built into the

architecture SHOULD be to preserve privacy instead of optimizing. The reason is that most users will not change defaults, and the default be one of privacy, only moving away from it by customer choice.

- o If possible, information about who is gathering data about a user SHOULD be available to that user. Everyone deserves to know who is watching them.
- o Proposals SHOULD address the issue of loss of geographic location privacy due to monitoring of packets crossing the Internet, and find an explicit balance between conflicting goals.
- o Protocols SHOULD avoid using identifiers for multiple purposes. Different identifier scopes do not need to overlap. Confidentiality boundaries can be established by clearly defining limited interfaces.
- o Protocols SHOULD avoid using long-lasting identifiers in scopes where they are not necessary.

## 6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

## 7. Security Considerations

In a sense this entire document is about security.

## 8. Acknowledgements

Thanks to many with whom we have discussed this issue in recent months.

## 9. Normative References

[I-D.ietf-mext-rfc3775bis]

Perkins, C., Johnson, D., and J. Arkko, "Mobility Support in IPv6", draft-ietf-mext-rfc3775bis-08 (work in progress), October 2010.



[I-D.irtf-rrg-recommendation]

Li, T., "Recommendation for a Routing Architecture",  
draft-irtf-rrg-recommendation-14 (work in progress),  
September 2010.

[RAND-EDPD]

Robinson, N., Graux, H., Botterman, M., and L. Valeri,  
"Review of the European Data Protection Directive",  
May 2009.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3693] Cuellar, J., Morris, J., Mulligan, D., Peterson, J., and  
J. Polk, "Geopriv Requirements", RFC 3693, February 2004.

#### Authors' Addresses

Scott Brim  
Cisco

Email: [scott.brim@gmail.com](mailto:scott.brim@gmail.com)

Marc Linsner  
Cisco

Email: [mlinsner@cisco.com](mailto:mlinsner@cisco.com)

Bryan McLaughlin  
Cisco

Email: [brmclaug@cisco.com](mailto:brmclaug@cisco.com)

Klaas Wierenga  
Cisco

Email: [kwiereng@cisco.com](mailto:kwiereng@cisco.com)



Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: April 15, 2011

R. Despres  
RD-IPtech  
B. Carpenter  
Univ. of Auckland  
S. Jiang  
Huawei Technologies Co., Ltd  
October 12, 2010

Native IPv6 Across NAT44 CPEs (6a44)  
draft-despres-softwire-6a44-01

Abstract

Most CPEs should soon be dual stack, but a large installed base of IPv4-only CPEs is likely to remain for several years. Also, with the IPv4 address shortage, more and more ISPs will assign private IPv4 addresses to their customers. The need for IPv6 connectivity therefore concerns hosts behind IPv4-only CPEs, including such CPEs that are assigned private addresses. The 6a44 mechanism specified in this document addresses this need, without limitations and operational complexities of Tunnel Brokers and Teredo to do the same.

6a44 is based on an address mapping and on a mechanism whereby suitably upgraded hosts behind a NAT may obtain IPv6 connectivity via a stateless 6a44 server function operated by their Internet Service Provider. With it, IPv6 traffic between two 6a44 hosts in a single site remains within the site. Except for IANA numbers that remain to be assigned, the specification is intended to be complete enough for running codes to be independently written and interwork.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 15, 2011.

## Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Applicability . . . . .	4
3. 6a44 IPv6 Address Format . . . . .	6
4. Address Mappings and Encapsulations . . . . .	8
5. MTU considerations . . . . .	10
6. Host Acquisition of IPv6 Addresses and their Lifetimes . . . . .	10
7. Security considerations . . . . .	13
8. IANA Considerations . . . . .	14
9. Acknowledgments . . . . .	14
10. References . . . . .	14
10.1. Normative References . . . . .	14
10.2. Informative References . . . . .	15
Authors' Addresses . . . . .	16

## 1. Introduction

Most CPEs (customer premise equipments) should soon be dual stack, but a large installed base of IPv4-only CPEs is likely to remain for several years. Also, with the IPv4 address shortage, more and more Internet service providers (ISPs) will assign private IPv4 addresses of [RFC1918] to their customers. The need for IPv6 connectivity therefore includes hosts behind IPv4-only CPEs, including such CPEs that have private addresses.

At the moment, there are two traversal techniques to address this need:

1. A configured tunnel (IPv6 in IPv4 or even IPv6 in UDP), involving a managed tunnel broker, e.g. [RFC3053], with which the user must register. Well known examples include deployments of the Hexago tool, and the SixXs collaboration. However, this approach does not scale well; it requires significant support effort and is really only suitable for "hobbyist" early adopters of IPv6.
2. Teredo [RFC4380]. This is an automatic UDP-based tunneling solution that relies on a Teredo server, and on Teredo relays willing to carry the traffic. Unfortunately experience shows that this is sometimes an unreliable process in practice, with clients sometimes believing that they have Teredo connectivity when in fact they don't, or alternatively with the Teredo server and relay being very remote from the client and causing extremely long latency for IPv6 packets. This leads to user frustration and even to advice from help desks to disable IPv6.

6a44 is based on an address mapping and on a mechanism whereby suitably upgraded hosts behind a NAT may obtain IPv6 connectivity via a stateless 6a44 server function operated by their Internet Service Provider.

To address this need without the mentioned limitations, 6a44 is based on an address mapping and on a mechanism whereby suitably upgraded hosts behind a NAT may obtain IPv6 connectivity via a stateless 6a44 server function operated by their ISP. It can apply even with ISPs that, due to the IPv4 address shortage, assign private addresses of [RFC1918] to their IPv4 customers (typically with prefix 10.0.0.0/8).

6a44 is only a transition technology. It will no longer have to be used when the number of IPv4-only CPEs becomes negligible.

Except for IANA numbers that remain to be assigned, the specification is intended to be complete enough for running codes to be independently written and interwork.

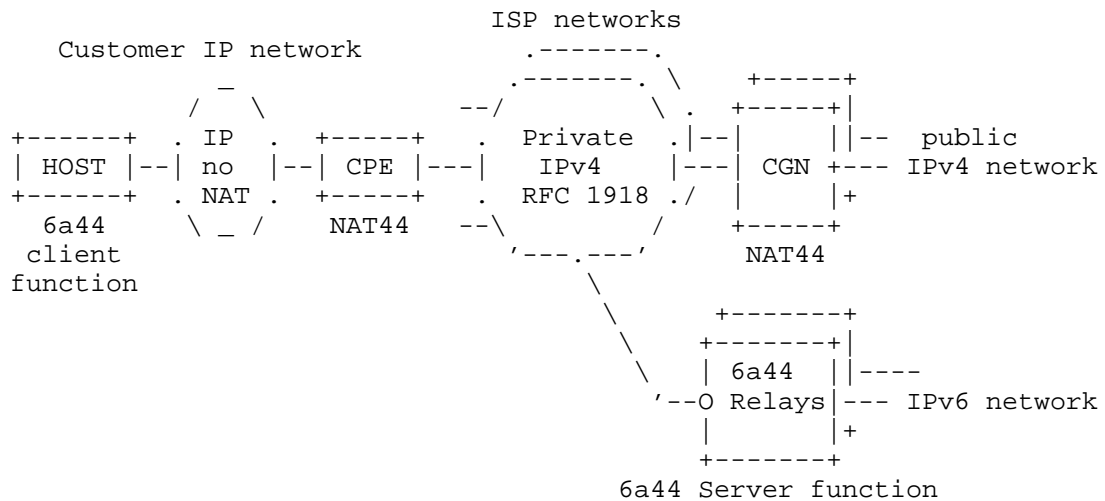
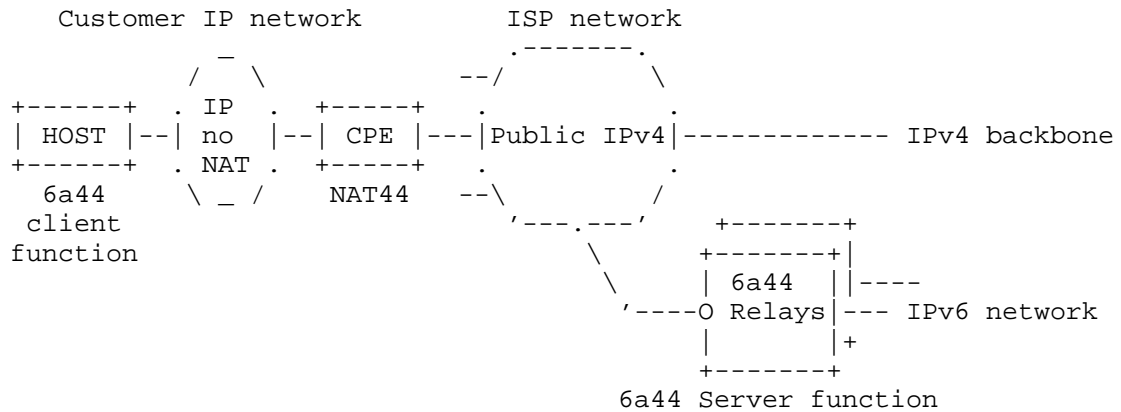
## 2. Applicability

Both hosts and ISPs can be made 6a44 capable independently of each other, with 6a44 being actually used by 6a44 capable hosts where their local ISPs are 6a44-capable.

For a host to be 6a44 capable, it has to support the 6a44 Client function ("6a44-C" in some Figures). This function is placed in its TCP/IP stack at the same place as the 6to4 router function of [RFC3056]: it has an IPv4 interface in its link-layer direction and both an IPv4 interface and an IPv6 pseudo-interface in its higher layers direction.

To enable its 6a44 function, a host must have no intra-site NAT44 between itself and the site CPE. (In sites where there are intra-site NAT44s, these NATs should be configured so that hosts behind them cannot enable 6a44. In view of the specification below, it can be done with a port mapping in each of them between the well-known port of 6a44 and an internal private address that DHCP doesn't assign.) In addition, the host must have in IPv4 a link MTU of at least 1308 octets (the MTU to be guaranteed in IPv6 plus the length of an UDP/IPv4 encapsulation header).

For an IPv4 ISP network to be 6a44 capable, the ISP must operate the 6a44 Server function, ("6a44-S" in some Figures). This function is anywhere at its border between the IPv4 network and an IPv6 network in which it has a /48 prefix. Typically this prefix will be chosen from whatever shorter PA prefix has been allocated to the ISP. The 6a44 server function can be replicated in any number of routers, known as "6a44 Relays", to enhance service quality and service availability. Also, the network must have an IPv4 MTU of at least 1308 octets and, for security, must support the ingress filtering of [RFC3704] (see Section 7).



#### 6a44 ISP CONFIGURATIONS

Figure 1

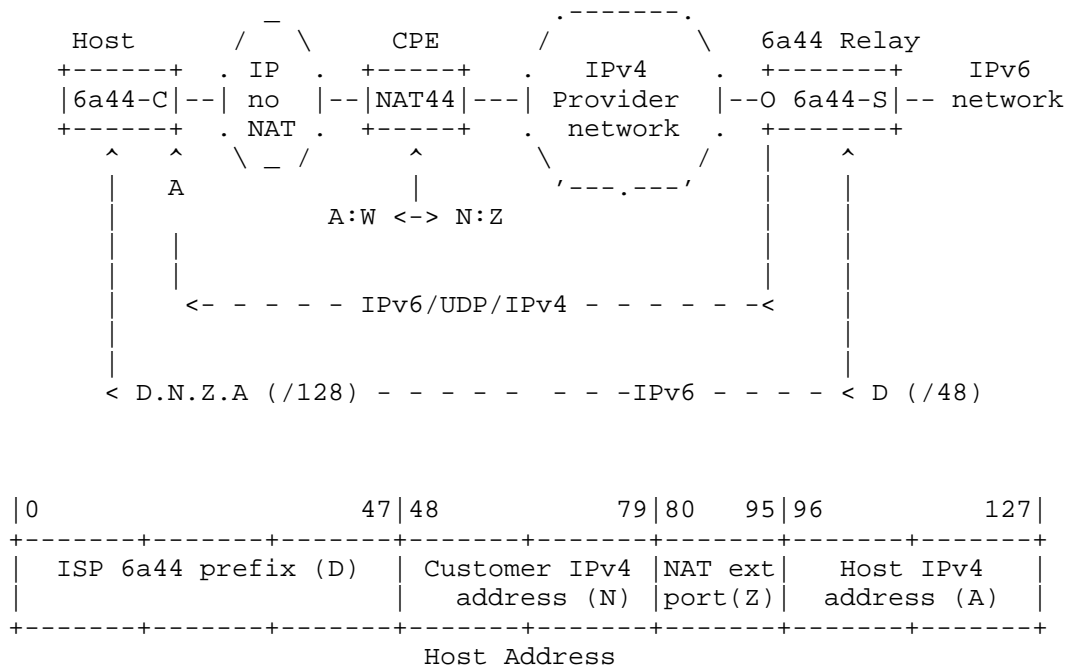
Each ISP can support one public-addressing and several private-addressing 6a44 networks.

In 6a44 networks, ISPs may route IPv6 in addition to IPv4. Where this is the case, 6a44 only concerns CPEs that are IPv4-only capable. If on the contrary IPv4 is the only routed address family, 6a44 may also concerns sites where CPEs are dual-stack capable. Unable to take advantage of their IPv6 capability, they act as if they would be IPv4-only.

Figure 1 illustrates ISP-network configurations on which 6a44 can be used.

NOTE: The objective of 6a44 differs from that of Teredo ([RFC4380] and [RFC5991]). Teredo has been designed to avoid needing any ISP participation. This has permitted early deployment but didn't ensure connectivity between all Teredo addresses and all native IPv6 addresses. Also, it imposed a very significant level of complexity. On the contrary, 6a44 is designed to be explicitly supported by ISPs. As a result, connectivity between 6a44 IPv6 addresses and all native IPv6 addresses can be ensured, and implementations can remain simple.

### 3. 6a44 IPv6 Address Format



HOST-ADDRESS CONSTRUCTION

Figure 2



The 6a44 IPv6 address an ISP assigns to a host must first contain all what is needed to reach it from the IPv6 backbone. This includes, as illustrated in Figure 2:

- o the IPv6 prefix D that the ISP has assigned border routers of its 6a44 network;
- o the IPv4 address N of the customer site (external address of the NAT44 in its CPE);
- o the port Z that, in the CPE NAT44 CPE, has to be used to reach the host at its address address A, and in the host the 6a44 well-known port W (to be assigned by IANA).

To ensure that two 6a44 hosts behind the same IPv4-only CPE exchange packets without entering the ISP network, the 6a44 address of each host must also contain its IPv4 address A.

The format of 6a44 IPv6 addresses, a concatenation of D,N,Z, and A, where D has to be a /48 prefix, is detailed in Figure 2.

NOTE: Since IPv6 prefixes D assigned by ISPs to their customers always start with 001, the prefix of all IPv6 Aggregatable Global Unicast addresses specified in [RFC2374], 6a44 IPv6 addresses bend the rule of [RFC4291] that says 'for all unicast addresses, except those that start with binary value 000, Interface IDs are required to be 64 bits long and to be constructed in Modified EUI-64 format'. This is however acceptable in practice because 6a44 addresses are never used on any real IPv6 link, and in particular never subject to the neighbor discovery protocol of [RFC2461] which depends on properties of interface IDs. A revision of the [RFC4291] sentence should eventually clarify this point.

#### 4. Address Mappings and Encapsulations

Figure 3 and Figure 4 detail the address mappings and encapsulations/decapsulations to be performed by 6a44 Client and server functions respectively, with the following notation:

- o (vX,A1,A2,data): a packet of the IPvX version that has A1 as source address, A2 as destination address, and "data" as payload.  
(UDP,P1,P2,data): a UDP IP payload that has P1 as source port, P2 as destination port, and "data" as payload.
- o B is the 6a44 well-known anycast address, that of the 6a44 Server function. X...: an address that starts with prefix X.
- o not X: an address different from X
- o X.Y: the concatenation of X and Y (the dot is the concatenation operator).

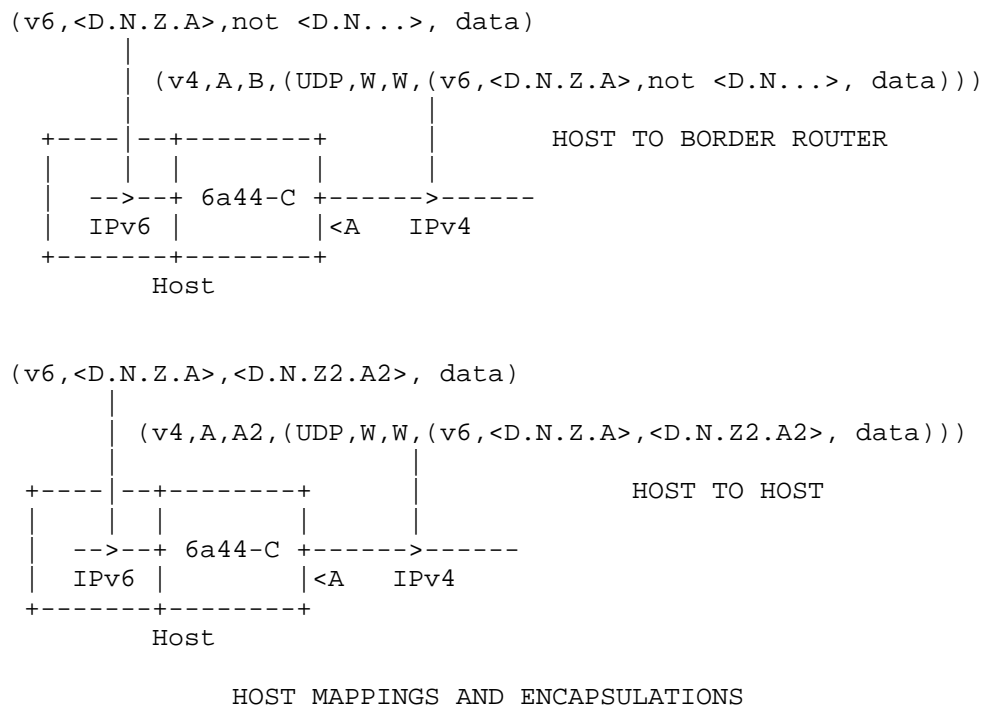


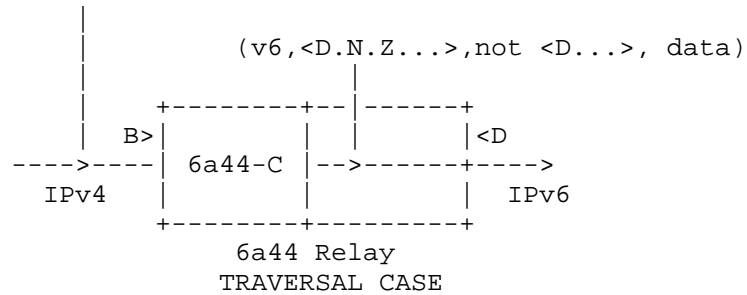
Figure 3

For protection against spoofing attacks, decapsulating functions must check consistency of IPv6 addresses fields with IPv4 addresses and UDP ports of encapsulating headers, both for source and destination addresses.

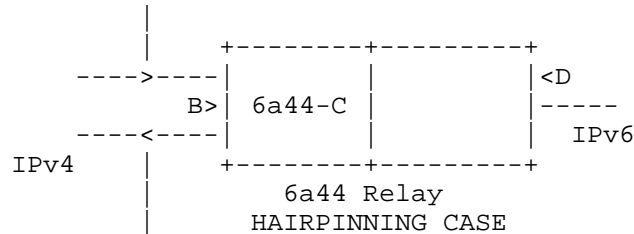
Figures present only one direction of 6a44-function traversals, but mappings that apply to the reverse direction are the same, with just a permutation of source and destination fields, for all of IPv4, IPv6, and UDP. Mappings and encapsulations/decapsulations for the reverse direction of that presented in Figures are the same, but with source and destination permuted in IPv6, IPv4 and UDP.

Recommendations of [RFC4213] that concern these encapsulations have to be followed.

$(v4, \langle N = \text{not } B \rangle, B, (\text{UDP}, Z, W, (v6, \langle D.N.Z \dots \rangle, \text{not } \langle D \dots \rangle, \text{data})))$



$(v4, \langle N1 = \text{not } B \rangle, B, (\text{UDP}, Z1, W, (v6, \langle D.N1.Z1 \dots \rangle, \langle D.N2.Z2 \dots \rangle, \text{data})))$



$(v4, B, N2, (\text{UDP}, B, Z2, (v6, \langle D.N1.Z1 \dots \rangle, \langle D.N2.Z2 \dots \rangle, \text{data})))$

#### 6a44-RELAY MAPPINGS AND ENCAPSULATIONS

Figure 4

## 5. MTU considerations

Reassembly of multi-fragment datagrams needs stateful processing, and opens the door to some denial of service attacks. To ensure a freedom of distribution of 6a44 Server functions in any number of parallel processors anywhere in 6a44 ISP networks, it has therefore to be avoided.

For this:

- o 6a44 ISP networks must have internal IPv4 MTUs of at least 1308 octets (which is easy to ensure).
- o 6a44 hosts must limit to 1280 octets IPv6 packets they transmit to destinations that are not neighbors on their own links. This behavior is already the normal one as long as no other IPv6 path MTU has been reliably discovered.
- o 6a44 Server functions refuse packets received from their IPv6 pseudo interfaces if their sizes exceed 1280 octets, with ICMPv6 Packet Too Big messages returned to sources as required by [RFC2460].)

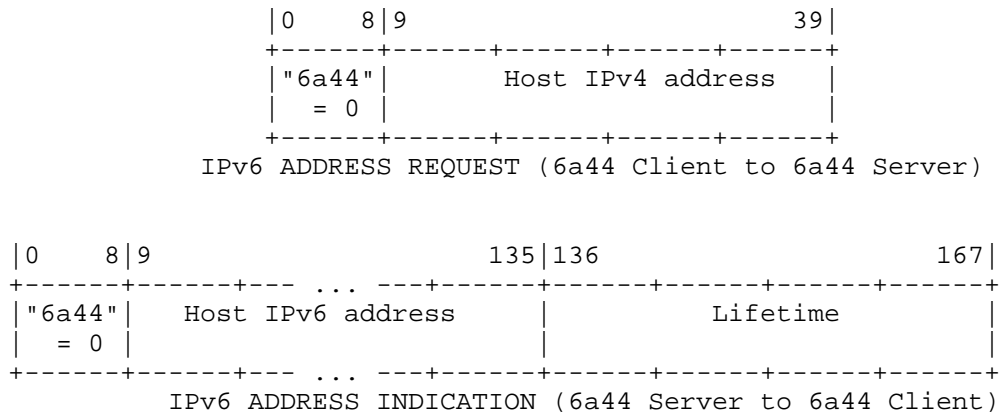
In a host, a destination is considered to be an on link neighbor if the IPv6 destination has the same bits 0-79 as the host address, and if the IPv4 destination starts with the prefix of the IPv4 link of the host. In this case, the IPv6 path MTU can be taken as that of the IPv4 link MTU minus 28 octets (a value that is typically significantly longer than 1280 octets).

## 6. Host Acquisition of IPv6 Addresses and their Lifetimes

Acquisition of 6a44 addresses by hosts is independent from other mechanisms they may have to acquire other IPv6 addresses (PPP, DHCP, SLAAC, ...). It only depends on 6a44 packet exchanges between hosts and 6a44 Relays.

In order to acquire 6a44 addresses, hosts transmit IPv6 Address Request messages to 6a44 Server functions and expect IPv6 Address Indication messages in return.

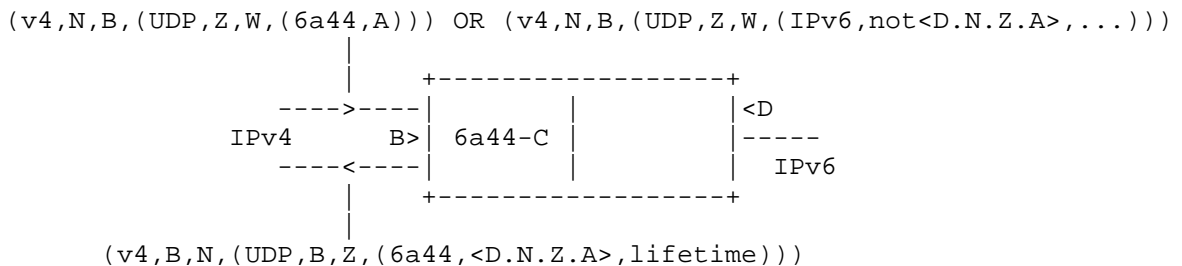
Formats of these 6a44 messages are shown in Figure 5. They start with a 6a44 mark, a null octet chosen so that, in payloads of UDP datagrams received by 6a44 Client and 6a44 Server functions, 6a44 messages can be distinguished from IPv6 packets (IPv6 packets always have a non-null first octet).



## 6a44 MESSAGES

Figure 5

Message processing in 6a44 Server function is shown in Figure 6 with the same notation as in Section 4. The lifetime of returned IPv6 addresses should be the same as that of IPv4 addresses assigned by the same ISP. it is expressed in seconds.



## 6a44 MESSAGE PROCESSING IN BORDER ROUTERS

Figure 6

In a host, the 6a44 Client function should be activated for one of its physical interfaces only if this interface has a private IPv4 address and no other native IPv6 address. (An address is said to be native if it starts with 2000::/3 (global unicast) and neither with 2002::/16 (the 6to4 prefix) nor with 2001::/32 (the Teredo prefix).)

Message processing in a 6a44 Client function consists in transmitting from time to time IPv6 Address Requests to the 6a44 Server function, and to update the host IPv6 address and its lifetime each time an IPv6 Address Indication message is received (with due IPv4 source address verification for security).

In order to decide when to transmit such a message, the 6a44 Client function has the equivalent of the following states:

"Waiting for an IPv6 Address Indication": When this state is entered, an IPv6 Address Request is transmitted, a Response Awaited timer of 1 second is started, and a Retransmission Count is set to 0. If the timer expires with a Retransmission Count less than 10, a new IPv6 Address Request is transmitted, and the count is increased by 1. If it expires with a count equal to 10, the state is changed to "waiting before a new attempt to find a 6a44 service". If an IPv6 Address Indication is received while in this state, the timer is stopped, the state is changed to "Waiting for having to refresh the NAT-binding". This state is also re-entered each time a new IPv4 address is assigned to the link-direction interface of the 6a44 Client function.

"Waiting for having to refresh the NAT-binding": When this state is entered, a timer of 29 second is started. (This value is that chosen for SIP in [RFC5626] for the same objective, i.e. to maintain tunnel NAT bindings without particular knowledge about NAT specifics.) This timer is restarted each time an IPv6 packet is transmitted to the 6a44 Server function (not when a packet is transmitted host to host within the customer site). It is also restarted if an IPv6 Address Indication is received while in this state. (This may happen in particular if the NAT binding has changed, e.g. because CPE reset during the lifetime of the IPv6 address.) If the timer expires, the state is changed to "Waiting for an IPv6 Address Indication".

"waiting before a new attempt to find a 6a44 service": When this state is entered, a 6a44 Availability timer of 1 hour is started. When it expires, the state is changed to "Waiting for an IPv6 Address Indication".

## 7. Security considerations

**Traffic-capture attack by a neighbor:** If it would be possible to transmit from a neighboring site a bogus address indication to a 6a44 host, this host could inadvertently advertise an IPv6 address that is not his real 6a44 address. Some incoming connections that it should have received could then be redirected to a wrong address. However, because 6a44 is applicable only to ISP networks that support the ingress filtering of [RFC3704] (see Section 2), no neighbor can fake a valid Address Indication message (the IPv4 source of packets it sends cannot be the 6a44 well-known IPv4 address, the only valid source for an Address Indication message).

**Spoofing attacks:** With address checks of Section 4, 6a44 should introduce no spoofing vulnerabilities beyond those the underlying IPv4 networks may have. ISPs that use subscriber authentications to secure IPv4 address assignments have the effect of this authentication automatically extended to 6a44 addresses (they include the assigned IPv4 addresses).

**Denial-of-service attacks:** Provided 6a44 Server functions are provisioned with enough processing power, which is facilitated by their being stateless, 6a44 is expected to introduce no denial of service vulnerabilities of its own.

**Subscriber authentication:** This is not provided as part of 6a44, because it is assumed to have occurred when the IPv4 address assignment was made.

**Routing-loop attacks:** A risk of routing-loop attacks has been identified for some encapsulation/decapsulation mechanisms [draft-ietf-v6ops-tunnel-loops-00]. It doesn't exist with 6a44 because:

- \* IPv4 packets entering a 6a44 Server function are not forwarded if they come from another instance of the 6a44 Server function itself, i.e. if the IPv4 source is the 6a44 well-known IPv4 address Section 4.
- \* The encapsulation header, which is based on UDP with a specific well-known port, cannot be confused with that of other encapsulation mechanisms (in particular those of IP in IP like those of 6to4, 6rd and ISATAP).

Missing 6a44 Server function: If a 6a44-capable host is client of an ISP that doesn't support 6a44, 6a44 IPv6 Address Request messages transmitted by the host will be forwarded to the Internet backbone, with the 6a44 well-known IPv4 address as destination. Since this address doesn't start with any prefix that the backbone routes toward ISP networks, these messages will be discarded before reaching any place where a fake 6a44 Server could have been malevolently placed. There is therefore no danger that 6a44 hosts could have their IPv6 traffic routed via 6a44 Server functions that would not belong to their local ISP (i.e. where they could be observed and acted upon without control).

## 8. IANA Considerations

For 6a44 to be used, both its IPv4 well-known address B and its well-known port W need to be assigned by IANA.

This assignment is necessary to validate the plug-an-play operation of 6a44 with independent implementations. Having it as quickly as possible (i.e. without waiting for all details of the specification to be agreed on), would be helpful for an early validation of the 6a44 plug-and-play operation.

## 9. Acknowledgments

This specification results from a convergence effort of authors of [draft-despres-softwire-6rdplus-00] and [draft-carpenter-softwire-sample-00]. Useful comments have been received about these earlier proposals or later, in particular from Pascal Thubert, Dan Wing, Yu Lee, Olivier Vautrin, Fred Templin, and Ole Troan. They have to be thanked for their contributions.

## 10. References

### 10.1. Normative References

- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.



## 10.2. Informative References

- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2374] Hinden, R. and S. Deering, "An IPv6 Aggregatable Global Unicast Address Format", RFC 2374, July 1998.
- [RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.
- [RFC3053] Durand, A., Fasano, P., Guardini, I., and D. Lento, "IPv6 Tunnel Broker", RFC 3053, January 2001.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.
- [RFC5626] Jennings, C., Mahy, R., and F. Audet, "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", RFC 5626, October 2009.
- [RFC5991] Thaler, D., Krishnan, S., and J. Hoagland, "Teredo Security Updates", RFC 5991, September 2010.
- [draft-carpenter-softwire-sample-00]  
Carpenter, B. and S. Jiang, "Legacy NAT Traversal for IPv6: Simple Address Mapping for Premises - Legacy Equipment (SAMPLE)", June 2010.
- [draft-despres-softwire-6rdplus-00]  
Despres, R., "Rapid Deployment of Native IPv6 Behind IPv4 NATs (6rd+)", July 2010.
- [draft-ietf-v6ops-tunnel-loops-00]  
Nakibly, G. and F. Templin, "Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations - Work in progress", September 2010.

Authors' Addresses

Remi Despres  
RD-IPtech  
3 rue du President Wilson  
Levallois,  
France

Email: remi.despres@free.fr

Brian Carpenter  
Department of Computer Science  
University of Auckland  
PB 92019  
Auckland, 1142  
New Zealand

Email: brian.e.carpenter@gmail.com

Sheng Jiang  
Huawei Technologies Co., Ltd  
KuiKe Building, No.9 Xinxu Rd.,  
Shang-Di Information Industry Base, Hai-Dian District, Beijing,  
P.R. China

Email: shengjiang@huawei.com



Internet Engineering Task Force  
Internet-Draft  
Intended status: BCP  
Expires: April 21, 2011

A. Durand  
Juniper Networks  
I. Gashinsky  
Yahoo! Inc.  
D. Lee  
Facebook, Inc.  
S. Sheppard  
ATT Labs  
October 18, 2010

Logging recommendations for Internet facing servers  
draft-durand-server-logging-recommendations-00

Abstract

In the wake of IPv4 exhaustion and deployment of IP address sharing techniques, this document recommends that Internet facing servers log port number and accurate timestamps in addition to the incoming IP address.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 21, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Recommendations . . . . .	3
3. ISP Considerations . . . . .	4
4. IANA Considerations . . . . .	4
5. Security Considerations . . . . .	4
6. References . . . . .	5
6.1. Normative references . . . . .	5
6.2. Informative references . . . . .	5
Authors' Addresses . . . . .	5

## 1. Introduction

According to the most recent predictions, the global IPv4 address free pool at IANA will exhaust sometime in 2011. After that, service providers will have a hard time finding enough IPv4 global addresses to sustain product and subscriber growth. Due to the huge global existing infrastructure, both hardware and software, vendors and service providers must continue to support IPv4 technologies for the foreseeable future. As legacy applications and hardware are retired the reliance on IPv4 will diminish but this is a years long perhaps decades long process.

To maintain legacy IPv4 address support, service providers will have little choice but to share IPv4 global addresses among multiple customers. Techniques to do so are outside of the scope of this documents. All include some form of address translation/address sharing, being NAT44, NAT64 or DS-Lite.

The effects on the Internet of the introduction of those address sharing techniques have been documented in [I-D.ietf-intarea-shared-addressing-issues].

Address sharing techniques come with their own logging infrastructure to track the relation between which original IP address and source port(s) were associated with which user and external IPv4 address at any given point in time. In the past to support abuse mitigation or public safety requests, the knowledge of the external global IP address was enough to identify a subscriber of interest. With address sharing technologies, only providing information about the external public address associated with a session to a service provider is no longer sufficient information to unambiguously identify customers.

Note: this document provides recommendations for Internet facing servers logging incoming connections. Its does not provide any recommendations about logging on carrier-grade NAT or other address sharing tools.

## 2. Recommendations

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

It is RECOMMENDED as best current practice that Internet facing servers logging incoming IP addresses also log:

- o The source port number.
- o A timestamp accurate to the second, with associated time zone.
- o The transport protocol (usually TCP or UDP) and destination port number, when the server application is defined to use multiple transports or multiple ports.

Discussion: Carrier-grade NATs may have different policies to recycle ports, some implementations may decide to reuse ports almost immediately, some may wait several minutes before marking the port ready for reuse. As a result, servers have no idea how fast the ports will be reused and, thus, should log timestamps using a reasonably accurate clock. At this point the RECOMMENDED accuracy for timestamps is to the second or better.

Examples of Internet facing servers include, but are not limited to, web servers and email servers.

Although the deployment of address sharing techniques is not immediately foreseen in IPv6, the above recommendations apply to both IPv4 and IPv6, if only for consistency and code simplification reasons.

Discussions about data retention policies are out of scope for this document.

### 3. ISP Considerations

ISP deploying IP address sharing techniques should also deploy a corresponding logging architecture to maintain records of the relation between customers identity and IP/port resources they utilize. However, recommendation on this topic are out of scope for this document.

### 4. IANA Considerations

None.

### 5. Security Considerations

In the absence of source port number and accurate timestamp, operators deploying any address sharing techniques will not be able to identify unambiguously customers when dealing with abuse or public safety queries.

## 6. References

### 6.1. Normative references

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

### 6.2. Informative references

[I-D.ietf-intarea-shared-addressing-issues]  
Ford, M., Boucadair, M., Durand, A., Levis, P., and P.  
Roberts, "Issues with IP Address Sharing",  
draft-ietf-intarea-shared-addressing-issues-02 (work in  
progress), October 2010.

## Authors' Addresses

Alain Durand  
Juniper Networks  
1194 North Mathilda Avenue  
Sunnyvale, CA 94089-1206  
USA

Email: [adurand@juniper.net](mailto:adurand@juniper.net)

Igor Gashinsky  
Yahoo! Inc.  
45 West 18th St.  
New York, NY 10011  
USA

Email: [igor@yahoo-inc.com](mailto:igor@yahoo-inc.com)

Donn Lee  
Facebook, Inc.  
1601 S. California Ave.  
Palo Alto, CA 94304  
USA

Email: [donn@facebook.com](mailto:donn@facebook.com)



Scott Sheppard  
ATT Labs  
575 Morosgo Ave, 4d57  
Atlanta, GA 30324  
USA

Email: [Scott.Sheppard@att.com](mailto:Scott.Sheppard@att.com)



Internet Area WG  
Internet Draft  
Updates: 791,1122,2003  
Intended status: Proposed Standard  
Expires: April 2011

J. Touch  
USC/ISI  
October 22, 2010

Updated Specification of the IPv4 ID Field  
draft-ietf-intarea-ipv4-id-update-01.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on April 22, 2011.

## Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Abstract

The IPv4 Identification (ID) field enables fragmentation and reassembly, and as currently specified is required to be unique within the maximum lifetime on all datagrams. If enforced, this uniqueness requirement would limit all connections to 6.4 Mbps. Because this is obviously not the case, it is clear that existing systems violate the current specification. This document updates the specification of the IPv4 ID field to more closely reflect current practice and to more closely match IPv6 so that the field is defined only when a datagram is actually fragmented. It also discusses the impact of these changes on how datagrams are used.

## Table of Contents

1. Introduction.....	3
2. Conventions used in this document.....	3
3. The IPv4 ID Field.....	3
4. Uses of the IPv4 ID Field.....	4
5. Background on IPv4 ID Reassembly Issues.....	5
6. Updates to the IPv4 ID Specification.....	6
6.1. IPv4 ID Used Only for Fragmentation.....	7
6.2. Encourage Safe IPv4 ID Use.....	8
6.3. IPv4 ID Requirements That Persist.....	9
7. Impact on Datagram Use.....	9
8. Updates to Existing Standards.....	10
8.1. Updates to RFC 791.....	10
8.2. Updates to RFC 1122.....	11
8.3. Updates to RFC 2003.....	11
9. Impact on NATs and Tunnel Ingresses.....	12
10. Impact on Header Compression.....	13

11. Security Considerations.....	13
12. IANA Considerations.....	13
13. References.....	14
13.1. Normative References.....	14
13.2. Informative References.....	14
14. Acknowledgments.....	15

## 1. Introduction

In IPv4, the Identification (ID) field is a 16-bit value that is unique for every datagram for a given source address, destination address, and protocol, such that it does not repeat within the Maximum Segment Lifetime (MSL) [RFC791][RFC1122]. As currently specified, all datagrams between a source and destination of a given protocol must have unique IPv4 ID values over a period of this MSL, which is typically interpreted as two minutes (120 seconds). This uniqueness is currently specified as for all datagrams, regardless of fragmentation settings.

The uniqueness of the IPv4 ID is a known problem for high speed devices; if strictly enforced, it would limit the speed of a single protocol between two endpoints to 6.4 Mbps for typical MTUs of 1500 bytes [RFC4963]. It is common for a single protocol to operate far in excess of these rates, which strongly indicates that the uniqueness of the IPv4 ID as specified is already moot.

This document updates the specification of the IPv4 ID field to more closely reflect current practice, and to include considerations taken into account during the specification of the similar field in IPv6.

## 2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, the characters ">>" proceeding an indented line(s) indicates a requirement using the key words listed above. This convention aids reviewers in quickly identifying or finding this document's explicit requirements.

## 3. The IPv4 ID Field

IP supports datagram fragmentation, where large datagrams are split into smaller components to traverse links with limited maximum transmission units (MTUs). Fragments are indicated in different ways in IPv4 and IPv6:

- o In IPv4, fragments are indicated using four fields of the basic header: Identification (ID), Fragment Offset, a "Don't Fragment" flag (DF), and a "More Fragments" flag (MF) [RFC791]
- o In IPv6, fragments are indicated in an extension header that includes an ID, Fragment Offset, and MF flag similar to their counterparts in IPv4 [RFC2460]

IPv4 and IPv6 fragmentation differs in a few important ways. IPv6 fragmentation occurs only at the source, so a DF bit is not needed to prevent downstream devices from initiating fragmentation (i.e., IPv6 always acts as if DF=1). The IPv6 fragment header is present only when a datagram has been fragmented, so the ID field is not present for non-fragmented datagrams, and thus is meaningful only for fragments. Finally, the IPv6 ID field is 32 bits, and required unique per source/destination address pair for IPv6, whereas for IPv4 it is only 16 bits and required unique per source/destination/protocol triple.

This document focuses on the IPv4 ID field issues, because in IPv6 the field is larger and present only in fragments.

#### 4. Uses of the IPv4 ID Field

The IPv4 ID field was originally intended for fragmentation and reassembly [RFC791]. Within a given source address, destination address, and protocol, fragments of an original datagram are matched based on their IPv4 ID. This requires that IDs are unique within the address/protocol triple when fragmentation is possible (e.g., DF=0) or when it has already occurred (e.g., frag\_offset>0 or MF=1).

The IPv4 ID field can be useful for other purposes. The field has been suggested as a way to detect and remove duplicate datagrams, e.g., at congested routers, although this has been noted and no current deployments are known (see Sec. 3.2.1.5 of [RFC1122]). It can similarly be used at end hosts to reduce the impact of duplication on higher-layer protocols (e.g., additional processing in TCP, or the need for application-layer duplicate suppression in UDP).

The IPv4 ID field can also be used to validate payloads of ICMP responses as matching the originally transmitted datagram at a host [RFC4963]. In this case, the ICMP payload - an IP datagram prefix - is matched against a cache of recently transmitted IP headers to check that the received ICMP reflects a transmitted datagram. At a tunnel ingress, the IPv4 ID enables returning ICMP messages to be matched to a cache of recently transmitted datagrams, to support ICMP relaying, with similar challenges [RFC2003].

Uses of the IPv4 ID field beyond fragmentation and reassembly require that the IPv4 ID be unique across all datagrams, not only when fragmentation is enabled. This document deprecates all such non-fragmentation uses.

## 5. Background on IPv4 ID Reassembly Issues

The following is a summary of issues with IPv4 fragment reassembly in high speed environments raised previously [RFC4963]. Readers are encouraged to consult RFC 4963 for a more detailed discussion of these issues.

With the maximum IPv4 datagram size of 64KB, a 16-bit ID field that does not repeat within 120 seconds means that the aggregate of all TCP connections of a given protocol between two endpoints is limited to roughly 286 Mbps; at a more typical MTU of 1500 bytes, this speed drops to 6.4 Mbps [RFC4963]. This limit currently applies for all IPv4 datagrams within a single protocol (i.e., the IPv4 protocol field) between two IP addresses, regardless of whether fragmentation is enabled or inhibited, and whether a datagram is fragmented or not.

IPv6, even at typical MTUs, is capable of 18.7 Tbps with fragmentation between two endpoints as an aggregate across all protocols, due to the larger 32-bit ID field (and the fact that the IPv6 next-header field, the equivalent of the IPv4 protocol field, is not considered in differentiating fragments). When fragmentation is not used the field is absent, and in that case IPv6 speeds are not limited by the ID field uniqueness.

Note also that 120 seconds is only an estimate on the maximum datagram lifetime. It is loosely based on half maximum value of the IP TTL field (255), measured in seconds, because the TTL is decremented not only for each hop, but also for each second a datagram is held at a router (as implied in [RFC791]). Network delays are incurred in other ways, e.g., satellite links, which can add seconds of delay even though the TTL is often not decremented by a corresponding amount. There is thus no enforcement mechanism to ensure that datagrams older than 120 seconds are discarded.

Wireless Internet devices are frequently connected at speeds over 54 Mbps, and wired links of 1 Gbps have been the default for several years. Although many end-to-end transport paths are congestion limited, these devices easily achieve 100+ Mbps application-layer throughput over LANs (e.g., disk-to-disk file transfer rates), and numerous throughput demonstrations have been performed with COTS systems over wide-area paths at these speeds for over a decade. This

strongly suggests that IPv4 ID uniqueness has been moot for a long time.

## 6. Updates to the IPv4 ID Specification

This document updates the specification of the IPv4 ID field in three distinct ways, as discussed in subsequent subsections:

- o Use the IPv4 ID field only for fragmentation
- o Avoiding a performance impact when the IPv4 ID field is used
- o Encourage safe operation when the IPv4 ID field is used

There are two kinds of datagrams used in the following discussion, named as follows:

- o Atomic datagrams: datagrams not yet having been fragmented (MF=0 and fragment offset=0) and for which further fragmentation has been inhibited (DF=1), i.e., as a C-code expression:

`(DF==1)&&(MF==0)&&(frag_offset==0)`

- o Non-atomic datagrams: datagrams which have either already been fragmented, i.e.:

`(MF=1) || (frag_offset>0)`

or for which fragmentation remains possible:

`(DF=0)`

I.e., non-atomic datagrams can be expressed in two equivalent tests:

`(DF==0) || (MF==1) || (frag_offset>0)`

which can also be expressed as follows, using DeMorgan's Law and other identities:

`~((DF==1)&&(MF==0)&&(frag_offset==0))`

Note that this final expression is the same as "not(atomic)".



### 6.1. IPv4 ID Used Only for Fragmentation

Although RFC1122 suggests the IPv4 ID field has other uses, this document asserts that this field is defined only for fragmentation and reassembly.

- o >> IPv4 ID field MUST NOT be used for purposes other than fragmentation and reassembly.

This has a few implications. In atomic datagrams, the IPv4 ID field has no meaning, and thus can be set to an arbitrary value, i.e., the requirement for non-repeating IDs within the address/protocol triple is no longer required for atomic datagrams:

- o >> Originating sources MAY set the IPv4 ID field of atomic datagrams to any value.

Second, all network nodes, whether at intermediate routers, destination hosts, or other devices (e.g., NATs, firewalls, tunnel egresses), cannot rely on the field:

- o >> All devices that examine IPv4 headers MUST ignore the IPv4 ID field of atomic datagrams.

The IPv4 ID field is thus meaningful only for non-atomic datagrams - datagrams that have either already been fragmented, or those for which fragmentation remains permitted. Atomic datagrams are detected by their DF, MF, and fragmentation offset fields as explained in Section 6, because such a test is completely backward compatible; this document thus does not reserve any IPv4 ID values, including 0, as distinguished.

Deprecating the use of the IPv4 ID field for non-reassembly uses should have little - if any - impact. IPv4 IDs are already frequently repeated, e.g., over even moderately fast connections. Duplicate suppression was only suggested [RFC1122], and no impacts of IPv4 ID reuse have been noted. Routers are not required to issue ICMPs on any particular timescale, and so IPv4 ID repetition should not have been used for validation, and again repetition occurs and probably could have been noticed [RFC1812]. ICMP relaying at tunnel ingresses is specified to use soft state rather than a datagram cache, and should have been noted if the latter for similar reasons [RFC2003].

## 6.2. Encourage Safe IPv4 ID Use

This document makes further changes to the specification of the IPv4 ID field and its use to encourage its safe use as corollary requirements changes as follows.

RFC 1122 discusses that TCP retransmits a segment it may be possible to reuse the IPv4 ID (see Section 8.2). This can make it difficult for a source to avoid IPv4 ID repetition for received fragments. RFC 1122 concludes that this behavior "is not useful"; this document formalizes that conclusion as follows:

- o >> The IPv4 ID of non-atomic datagrams MUST NOT be reused when sending a copy of an earlier non-atomic datagram.

RFC 1122 also suggests that fragments can overlap [RFC1122]. Such overlap can occur if successive retransmissions are fragmented in different ways but the same reassembly IPv4 ID.

This overlap is noted as the result of reusing IPv4 IDs when retransmitting datagrams, which this document deprecates. Overlapping fragments are themselves a hazard [RFC4963]. As a result:

- o >> Overlapping datagrams MUST be silently ignored during reassembly.

The IPv4 ID of non-atomic datagrams also needs to remain stable, to ensure that existing fragments are not reassembled incorrectly, as well as to ensure that the uniqueness of the IDs as generated by the source is not undermined.

For atomic datagrams, because the IPv4 ID field is ignored on receipt, it can be possible to rewrite the field. Rewriting can be useful to prevent use of the field as a covert channel, or to enable more efficient header compression. However, the IPv4 ID field needs to remain immutable when it is validated by higher layer protocols, such as IPsec. As a result:

- o >> The IPv4 ID field of non-atomic datagrams, or protected atomic datagrams MUST NOT change in transit; the IPv4 ID field of unprotected atomic datagrams MAY be changed in transit.

Protected datagrams are defined as those whose header fields are covered by integrity validation, such as IPsec AH [RFC4302].

### 6.3. IPv4 ID Requirements That Persist

This document does not relax the IPv4 ID field uniqueness requirements of [RFC791] for non-atomic datagrams, i.e.:

- o >> Sources emitting non-atomic datagrams MUST NOT repeat IPv4 ID values within one MSL for a given source address/destination address/protocol triple.

Such sources include originating hosts, tunnel ingresses, and NATs (see Section 9).

This document does not relax the requirement that all network devices honor the DF bit, i.e.:

- o >> IPv4 datagrams whose DF=1 MUST NOT be fragmented.
- o >> IPv4 datagram transit devices MUST NOT clear the DF bit.

In specific, DF=1 prevents fragmenting datagrams that are integral. DF=1 also prevents further fragmenting received fragments. Fragmentation, either of an unfragmented datagram or of fragments, is current permitted only where DF=0 in the original emitted datagram, and this document does not change that requirement.

### 7. Impact on Datagram Use

The following is a summary of the recommendations that are the result of the previous changes to the IPv4 ID field specification.

Because atomic datagrams can use arbitrary IPv4 ID values, the ID field no longer imposes a performance impact in those cases. However, the performance impact remains for non-atomic datagrams. As a result:

- o >> Sources of non-atomic IPv4 datagrams MUST rate-limit their output to comply with the ID uniqueness requirements.

Such sources include, in particular, DNS over UDP [RFC2671].

Because there is no strict definition of the MSL, reassembly hazards exist regardless of the IPv4 ID reuse interval or the reassembly timeout. As a result:

- o >> Higher layer protocols SHOULD verify the integrity of IPv4 datagrams, e.g., using a checksum or hash that can detect reassembly errors (the UDP checksum is weak in this regard, but better than nothing), as in SEAL [RFC5320].

Additional integrity checks can be employed using tunnels, as in SEAL, IPsec, or SCTP [RFC4301][RFC4960][RFC5320]. Such checks can avoid the reassembly hazards that can occur when using UDP and TCP checksums [RFC4963], or when using partial checksums as in UDP-Lite [RFC3828]. Because such integrity checks can avoid the impact of reassembly errors:

- o >> Sources of non-atomic IPv4 datagrams using strong integrity checks MAY reuse the ID within MSL values smaller than is typical.

Note, however, that such more frequent reuse can still result in corrupted reassembly and poor throughput, although it would not propagate reassembly errors to higher layer protocols.

## 8. Updates to Existing Standards

The following sections address the specific changes to existing protocols indicated by this document.

### 8.1. Updates to RFC 791

RFC 791 states that:

The originating protocol module of an internet datagram sets the identification field to a value that must be unique for that source-destination pair and protocol for the time the datagram will be active in the internet system.

And later that:

Thus, the sender must choose the Identifier to be unique for this source, destination pair and protocol for the time the datagram (or any fragment of it) could be alive in the internet.

It seems then that a sending protocol module needs to keep a table of Identifiers, one entry for each destination it has communicated with in the last maximum datagram lifetime for the internet.

However, since the Identifier field allows 65,536 different values, some host may be able to simply use unique identifiers independent of destination.

It is appropriate for some higher level protocols to choose the identifier. For example, TCP protocol modules may retransmit an identical TCP segment, and the probability for correct reception would be enhanced if the retransmission carried the same

identifier as the original transmission since fragments of either datagram could be used to construct a correct TCP segment.

This document changes RFC 791 as follows:

- o IPv4 ID uniqueness applies to only non-atomic datagrams.
- o Non-atomic IPv4 datagrams retransmitted by higher level protocols are no longer permitted to reuse the ID value.

## 8.2. Updates to RFC 1122

RFC 1122 states that:

### 3.2.1.5 Identification: RFC-791 Section 3.2

When sending an identical copy of an earlier datagram, a host MAY optionally retain the same Identification field in the copy.

#### DISCUSSION:

Some Internet protocol experts have maintained that when a host sends an identical copy of an earlier datagram, the new copy should contain the same Identification value as the original. There are two suggested advantages: (1) if the datagrams are fragmented and some of the fragments are lost, the receiver may be able to reconstruct a complete datagram from fragments of the original and the copies; (2) a congested gateway might use the IP Identification field (and Fragment Offset) to discard duplicate datagrams from the queue.

This document changes RFC 1122 as follows:

- o The IPv4 ID field is no longer permitted for duplicate detection.
- o The IPv4 ID field is no longer repeatable for higher level protocol retransmission.
- o IPv4 datagram fragments no longer are permitted to overlap.

## 8.3. Updates to RFC 2003

This document updates how IPv4-in-IPv4 tunnels create IPv4 ID values for the IPv4 outer header [RFC2003], but only in the same way as for any other IPv4 datagram source.

## 9. Impact on NATs and Tunnel Ingresses

Network address translators (NATs) and address/port translators (NAPTs) rewrite IP fields, and tunnel ingresses (using IPv4 encapsulation) copy and modify some IPv4 fields, so all are considered sources, as do any devices that rewrite any portion of the source address, destination address, protocol, and ID tuple for non-atomic datagrams [RFC3022]. As a result, they are subject to all the requirements of any source, as has been noted.

NATs present a particularly challenging situation for fragmentation. Because NATs overwrite portions of the reassembly tuple in both directions, they can destroy tuple uniqueness and result in a reassembly hazard. Whenever IPv4 source address, destination address, or protocol fields are modified, a NAT needs to ensure that the ID field is generated appropriately, rather than simply copied from the incoming datagram. In specific:

- o >> NATs MUST ensure that the IPv4 ID field of datagrams whose address or protocol are translated comply with requirements as if the datagram were sourced by the NAT.

This compliance means that the IPv4 ID field of non-atomic datagrams translated at a NAT need to obey the uniqueness requirements of any IPv4 datagram source. Unfortunately, fragments already violate that requirement, as they repeat an IPv4 ID within the MSL for a given source address, destination address, and protocol triple.

Such problems with transmitting fragments through NATs are already known; translation is based on the transport port number, which is present in only the first fragment anyway [RFC3022]. This document underscores the point that not only is reassembly (and possibly subsequent fragmentation) required for translation, it can be used to avoid issues with IPv4 ID uniqueness.

Note that NATs/NAPTs already need to exercise special care when emitting datagrams on their public side, because merging datagrams from many sources onto a single outgoing source address can result in IPv4 ID collisions. This situation precedes this document, and is not affected by it. It is exacerbated in large-scale, so-called "carrier grade" NATs [Ni09].

Tunnel ingresses act as sources for the outermost header, but tunnels act as routers for the inner headers (i.e., the datagram as arriving at the tunnel ingress). Ingresses can fragment as originating sources of the outer header, because they control the uniqueness of that IPv4 ID field. They need to avoid fragmenting the datagram at the inner

header, for the same reasons as any intermediate device, as noted elsewhere in this document.

#### 10. Impact on Header Compression

Header compression algorithms already accommodate various ways in which the IPv4 ID changes between sequential datagrams. Such algorithms currently need to preserve the IPv4 ID.

When compression can assume a nonchanging IPv4 ID, efficiency can be increased. However, when compression assumes a changing ID as a default, having a non-changing ID can make compression less efficient (see footnote 21 of [RFC1144], which is optimized for non-atomic datagrams). This document thus does not recommend whether atomic IPv4 datagrams should use nonchanging or changing IDs, but rather allows those IDs to be modified in transit (as per Sec. 6.2), which can be used to accommodate more efficient compression as desired.

#### 11. Security Considerations

This document attempts to address the security considerations associated with fragmentation in IPv4 [RFC4459].

When the IPv4 ID is ignored on receipt (e.g., for atomic datagrams), its value becomes unconstrained; that field then can more easily be used as a covert channel. For some atomic datagrams - notably those not protected by IPsec Authentication Header (AH) [RFC4302] - it is now possible, and may be desirable, to rewrite the IPv4 ID field to avoid its use as such a channel.

The IPv4 ID also now adds much less entropy of the header of a datagram. The IPv4 ID had previously been unique (for a given source/address pair, and protocol field) within one MSL, although this requirement was not enforced and clearly is typically ignored. IDs of non-atomic datagrams are now required unique only within the expected reordering of fragments, which could substantially reduce the amount of entropy in that field. The IPv4 ID of atomic datagrams is not required unique, and so contributes no entropy to the header.

The deprecation of the IPv4 ID field's uniqueness for atomic datagrams can defeat the ability to count devices behind a NAT [Be02]. This is not intended as a security feature, however.

#### 12. IANA Considerations

There are no IANA considerations in this document.

The RFC Editor should remove this section prior to publication

### 13. References

#### 13.1. Normative References

- [RFC791] Postel, J., "Internet Protocol", RFC 791 / STD 5, September 1981.
- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", RFC 1122 / STD 3, October 1989.
- [RFC1812] Baker, F. (Ed.), "Requirements for IP Version 4 Routers", RFC 1812 / STD 4, Jun. 1995.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 / BCP 14, March 1997.
- [RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.

#### 13.2. Informative References

- [Be02] Bellovin, S., "A Technique for Counting NATted Hosts", Internet Measurement Conference, Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement, November 2002.
- [Ni09] Nishitani, T., I. Yamagata, S. Miyakawa, A. Nakagawa, H. Ashida, "Common Functions of Large Scale NAT (LSN) ", (work in progress), draft-nishitani-cgn-05, July 2010.
- [RFC1144] Jacobson, V., "Compressing TCP/IP Headers", RFC 1144, Feb. 1990.
- [RFC2460] Deering, S., R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, August 1999.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [RFC3828] Larzon, L-A., M. Degermark, S. Pink, L-E. Jonsson, Ed., G. Fairhurst, Ed., "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, July 2004.



- [RFC4301] Kent, S., K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, Dec. 2005.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, Dec. 2005.
- [RFC4459] Savola, P., "MTU and Fragmentation Issues with In-the-Network Tunneling", RFC 4459, April 2006.
- [RFC4960] Stewart, R. (Ed.), "Stream Control Transmission Protocol", RFC 4960, Sep. 2007.
- [RFC4963] Heffner, J., M. Mathis, B. Chandler, "IPv4 Reassembly Errors at High Data Rates," RFC 4963, July 2007.
- [RFC5320] Templin, F., Ed., "The Subnetwork Encapsulation and Adaptation Layer (SEAL)", RFC 5320, Feb. 2010.

#### 14. Acknowledgments

This document was inspired by of numerous discussions among the authors, Jari Arkko, Lars Eggert, Dino Farinacci, and Fred Templin, as well as members participating in the Internet Area Working Group. Detailed feedback was provided by Carlos Pignataro and Gorrry Fairhurst. This document originated as an Independent Stream draft co-authored by Matt Mathis, PSC, and his contributions are greatly appreciated.

This document was prepared using 2-Word-v2.0.template.dot.

#### Author's Address

Joe Touch  
USC/ISI  
4676 Admiralty Way  
Marina del Rey, CA 90292-6695  
U.S.A.

Phone: +1 (310) 448-9151  
Email: touch@isi.edu



Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: April 25, 2011

J. Livingood  
Comcast  
October 22, 2010

IPv6 AAAA DNS Whitelisting Implications  
draft-livingood-dns-whitelisting-implications-01

Abstract

The objective of this document is to describe what whitelisting of DNS AAAA resource records is, or DNS whitelisting for short, as well as what the implications of this emerging practice are and what alternatives may exist. The audience for this document is the Internet community generally, including the IETF and IPv6 implementers.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 25, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

# Table of Contents

1. Introduction . . . . .	4
2. How DNS Whitelisting Works . . . . .	5
3. Concerns Regarding DNS Whitelisting . . . . .	7
4. Similarities to Split DNS . . . . .	9
5. Likely Deployment Scenarios . . . . .	10
5.1. Deploying DNS Whitelisting Universally . . . . .	10
5.2. Deploying DNS Whitelisting On An Ad Hoc Basis . . . . .	11
6. What Problems Are DNS Whitelisting Implementers Trying To Solve? . . . . .	11
7. Implications of DNS Whitelisting . . . . .	12
7.1. Architectural Implications . . . . .	12
7.2. Public IPv6 Address Reachability Implications . . . . .	13
7.3. Operational Implications . . . . .	13
7.3.1. De-Whitelisting May Occur . . . . .	13
7.3.2. Authoritative DNS Server Operational Implications . . . . .	13
7.3.3. DNS Recursive Resolver Server Operational Implications . . . . .	14
7.3.4. Monitoring Implications . . . . .	15
7.3.5. Troubleshooting Implications . . . . .	15
7.3.6. Additional Implications If Deployed On An Ad Hoc Basis . . . . .	16
7.4. Homogeneity May Be Encouraged . . . . .	16
7.5. Technology Policy Implications . . . . .	17
7.6. IPv6 Adoption Implications . . . . .	18
8. Solutions . . . . .	18
8.1. Implement DNS Whitelisting Universally . . . . .	18
8.2. Implement DNS Whitelisting On An Ad Hoc Basis . . . . .	18
8.3. Do Not Implement DNS Whitelisting . . . . .	19
8.3.1. Solving Current End User IPv6 Impairments . . . . .	19
9. Security Considerations . . . . .	19
9.1. DNSSEC Considerations . . . . .	20
9.2. Authoritative DNS Response Consistency Considerations . . . . .	20
10. IANA Considerations . . . . .	20
11. Contributors . . . . .	20
12. Acknowledgements . . . . .	21
13. References . . . . .	21
13.1. Normative References . . . . .	21
13.2. Informative References . . . . .	22
Appendix A. Document Change Log . . . . .	22
Appendix B. Open Issues . . . . .	23
Author's Address . . . . .	23

## 1. Introduction

[EDITORIAL: This is a rough first -00 draft. Some sections have not yet been completed but will be soon. Suggestions on all parts of this document are eagerly solicited.]

This document describes the emerging practice of whitelisting of DNS AAAA resource records (RRs), or DNS whitelisting for short. It also explores the implications of this emerging practice and what alternatives may exist.

The practice of DNS whitelisting appears to have first been used by major web content sites. These web site operators observed that when they added AAAA RRs to their authoritative DNS servers that a small fraction of end users had slow or otherwise impaired access to a given web site with both AAAA and A RRs. The fraction of users with such impaired access has been estimated to be roughly 0.078% of total Internet users [IETF 77 DNSOP WG Presentation] [Network World Article on IETF 77 DNSOP WG Presentation]. Thus, in an example Internet Service Provider (ISP) network of 10 million users, approximately 7,800 of those users may experience such impaired access.

As a result of this impairment affecting end users of a given domain, a few large web site operators have begun to either implement DNS whitelisting or strongly consider the implementation of DNS whitelisting [Network World Article on DNS Whitelisting]. When implemented, DNS whitelisting in practice means that a domain's authoritative DNS will return a AAAA RR to DNS recursive resolvers [RFC1035] on the whitelist, while returning no AAAA RRs to DNS resolvers which are not on the whitelist. It is important to note that these web site operators are motivated to maintain a high-quality user experience for all of their users, and that they are attempting to shield users with impaired access from the symptoms of these impairments that would negatively affect their access to certain websites and related Internet resources.

[EDITORIAL: change web site operators --> domain operators?]

However, critics of this emerging practice of DNS whitelisting have articulated several concerns. Among these are that this is a very different behavior from the current practice concerning the publishing of IPv4 address records, that it may create a two-tiered Internet, that policies concerning whitelisting and de-whitelisting are opaque, that DNS whitelisting reduces interest in the deployment of IPv6, that new operational and management burdens are created, and that the costs and negative implications of DNS whitelisting outweigh the perceived benefits as compared to fixing underlying impairments.

This document explores the reasons and motivations for DNS whitelisting. It also explores the concerns regarding this emerging practice. As a result, readers can hopefully better understand what DNS whitelisting is, why some parties are implementing it, and why other parties are critical of the practice.

## 2. How DNS Whitelisting Works

DNS whitelisting is implemented in authoritative DNS servers, where those servers implement IP address-based restrictions on AAAA query responses, which contain IPv6 addresses. In practice DNS whitelisting has been primarily implemented by web server operators. For a given operator of the website `www.example.com`, that operator essentially applies an access control list (ACL) on their authoritative DNS servers, which are authoritative for the domain `example.com`. The ACL is then configured with the IPv4 and/or IPv6 addresses of DNS recursive resolvers on the Internet, which have been authorized to be added to the ACL and to therefore receive AAAA RR responses. These DNS recursive resolvers are operated by other parties, such as ISPs, universities, governments, businesses, individual end users, etc. If a DNS recursive resolver IS NOT on the ACL, then NO AAAA RRs with IPv6 addresses will be sent in response to a query for a given hostname in the `example.com` domain. However, if a DNS recursive resolver IS on the ACL, then AAAA RRs with IPv6 addresses will be sent in response to a query for a given hostname in the `example.com` domain.

In practice this generally means that a very small fraction of the DNS recursive resolvers on the Internet can receive AAAA responses with IPv6 addresses, which means that the large majority of DNS resolvers on the Internet will receive only A RRs with IPv4 addresses. Thus, quite simply, the authoritative server hands out different answers depending upon who is asking; with IPv4 and IPv6 records for some on the authorized whitelist, and only IPv4 records for everyone else. See Figure 1 and Figure 2 for two different visual descriptions of how this works in practice.

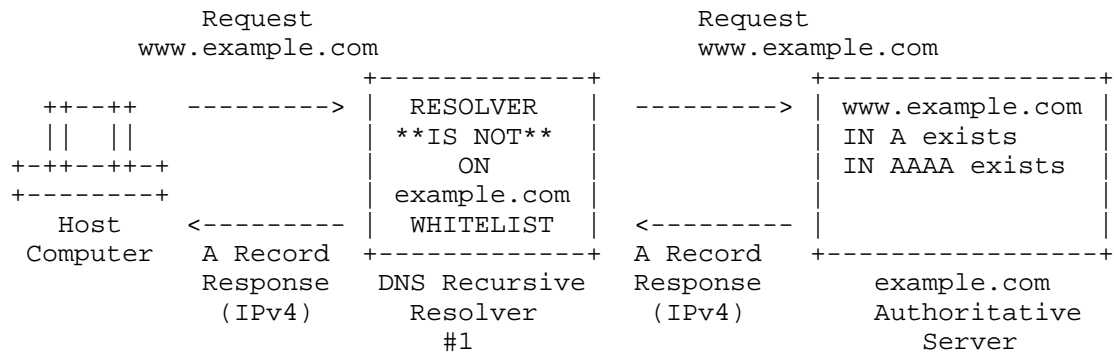
Finally, DNS whitelisting can be deployed in two primary ways: universally on a global basis, or on an ad hoc basis. These two potential deployment models are described in Section 5.

- 1: The authoritative DNS server for example.com receives a DNS query for www.example.com, for which both A (IPv4) and AAAA (IPv6) address records exist.
- 2: The authoritative DNS server examines the IP address of the DNS recursive resolver sending the query.
- 3: The authoritative DNS server checks this IP address against the access control list (ACL) that is the DNS whitelist.
- 4: If the DNS recursive resolver's IP address IS listed in the ACL, then the response to that specific DNS recursive resolver can contain both A (IPv4) and AAAA (IPv6) address records.
- 5: If the DNS recursive resolver's IP address IS NOT listed in the ACL, then the response to that specific DNS recursive resolver can contain only A (IPv4) address records and therefore cannot contain AAAA (IPv6) address records.

Figure 1: DNS Whitelisting - System Logic



-----  
A query is sent from a DNS recursive resolver that IS NOT on the DNS whitelist:



-----  
A query is sent from a DNS recursive resolver that IS on the DNS whitelist:

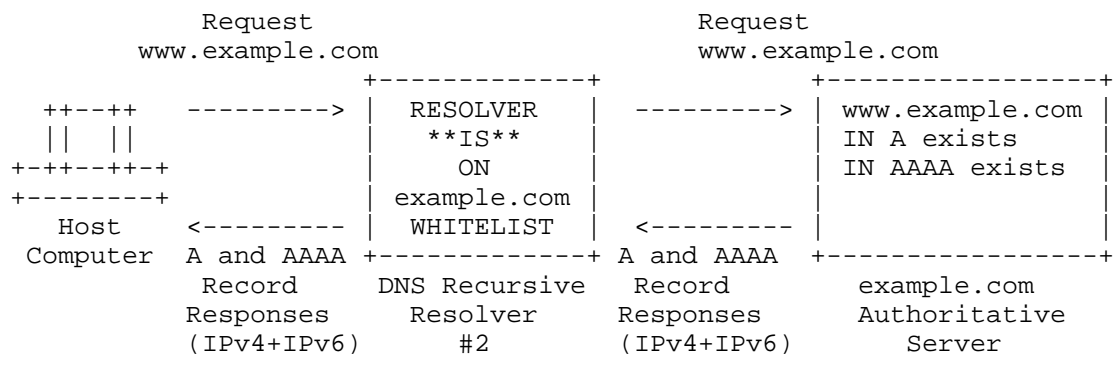


Figure 2: DNS Whitelisting - Functional Diagram

### 3. Concerns Regarding DNS Whitelisting

There are a number of potential implications relating to DNS whitelisting, which have raised various concerns in some parts of the Internet community. Many of those potential implications are described in Section 7.

Some parties in the Internet community are concerned that this emerging practice of DNS whitelisting for IPv6 address records could represent a departure from the generally accepted practices regarding IPv4 address records in the DNS on the Internet. These parties

explain their belief that for A records, containing IPv4 addresses, once an authoritative server operator adds the A record to the DNS, then any DNS recursive resolver on the Internet can receive that A record in response to a query. By extension, this means that any of the hosts connected to any of these DNS recursive resolvers can receive the IPv4 address records for a given FQDN. This enables new server hosts which are connected to the Internet, and for which a fully qualified domain name (FQDN) such as `www.example.com` has been added to the DNS with an IPv4 address record, to be almost immediately reachable by any host on the Internet. In this case, these new servers hosts become more and more widely accessible as new networks and new end user hosts connect to the Internet over time [EDITORIAL: consider reference to network effects]. It also means that the new server hosts do not need to know about these new networks and new end user hosts in order to make their content and applications available to them, in essence that each end in this end-to-end model is responsible for connecting to the Internet and once they have done so they can connect to each other without additional impediments or middle networks or intervening networks or servers knowing about these end points and whether one is allowed to contact the other.

In contrast, these parties are concerned that DNS whitelisting may fundamentally change this model. As a result, in this altered end-to-end model, one end (where the end user is located) cannot readily connect to the other end (where the content is located), without parts of the middle used by one end being known by the other end and approved for access to that end. Thus, as new networks connect to the Internet over time, those networks need to contact any and all domains which have implemented DNS whitelisting in order to apply to be added to their DNS whitelist, in the hopes of making the content and applications residing on named server hosts in those domains accessible by the end user hosts on that new network. Furthermore, this same need to contact all domains implementing DNS whitelisting also applies to all existing networks connected to the Internet.

Therefore, these concerned parties explain, whereas in the current IPv4 Internet when a new server host is added to the Internet it is widely available to all end user hosts and networks, when DNS whitelisting of IPv6 records is used then these new server hosts are not accessible to any end user hosts or networks until such time as the operator of the authoritative DNS servers for those new server hosts expressly authorizes access to those new server hosts by adding DNS recursive resolvers around the Internet to the ACL. This could represent a significant change in reachability of content and applications by end users and networks as these end user hosts and networks transition to IPv6. Therefore, a concern expressed is that if much of the content that end users are most interested in is not

accessible as a result, then end users and/or networks may resist adoption of IPv6 or actively seek alternatives to it, such as using multi-layer network address translation (NAT) techniques like NAT444 [I-D.shirasaki-nat444] on a long-term basis. There is also concern that this practice also could disrupt the continued increase in Internet adoption by end users if they cannot simply access new content and applications but must instead contact the operator of their DNS recursive resolver, such as their ISP or another third party, to have their DNS recursive resolver authorized for access to the content or applications that interests them. Meanwhile, these parties say, over 99.9% of all other end users that are also using that same network or DNS recursive resolver are unable to access the IPv6-based content, despite their experience being a positive one.

[EDITORIAL: Are there additional concerns to add here?]

#### 4. Similarities to Split DNS

DNS whitelisting as described herein is in some ways similar to so-called split DNS, which is briefly described in Section 3.8 of [RFC2775]. When split DNS is used, the authoritative DNS server returns different responses depending upon what host has sent the query. While [RFC2775] notes the typical use of split DNS is to provide one answer to hosts on an Intranet and a different answer to hosts on the Internet, the essence is that different answers are provided to hosts on different networks. This is basically the way that DNS whitelisting works, in so far as hosts of different networks, which use different DNS recursive resolvers, receive different answers if one DNS recursive resolver is on the whitelist and the other is not. Thus, in a way, DNS whitelisting could in some ways be considered split DNS on the public Internet, though with some differences.

In [RFC2956], Internet transparency and Internet fragmentation concerns regarding split DNS are detailed in Section 2.1. [RFC2956] further notes in Section 2.7, concerns regarding split DNS and that it "makes the use of Fully Qualified Domain Names (FQDNs) as endpoint identifiers more complex." Section 3.5 of [RFC2956] further recommends that maintaining a stable approach to DNS operations is key during transitions such as the one to IPv6 that is underway now, stating that "Operational stability of DNS is paramount, especially during a transition of the network layer, and both IPv6 and some network address translation techniques place a heavier burden on DNS."

## 5. Likely Deployment Scenarios

In considering how DNS whitelisting may emerge more widely, there are two likely deployment scenarios, which are explored below.

### 5.1. Deploying DNS Whitelisting Universally

The least likely deployment scenario is one where DNS whitelisting becomes a standardized process across all authoritative DNS servers, across the entire Internet. While this scenario is the least likely, due to some parties not sharing the concerns that have so far motivated the use of DNS whitelisting, it is nonetheless conceivable that it could be one of the ways in which DNS whitelisting may be deployed.

In order for this deployment scenario to occur, it is likely that DNS whitelisting functionality would need to be built into all authoritative DNS server software, and that all operators of authoritative DNS servers would have to upgrade their software and enable this functionality. Furthermore, it is likely that new Internet Draft documents would need to be developed which describe how to properly configure, deploy, and maintain DNS whitelisting. As a result, it is unlikely that DNS whitelisting would, at least in the next several years, become universally deployed. Furthermore, these DNS whitelists are likely to vary on a domain-by-domain basis, depending upon a variety of factors. Such factors may include the motivation of each domain owner, the location of the DNS recursive resolvers in relation to the source content, as well as various other parameters that may be transitory in nature, or unique to a specific end user host type. Thus, it is probably unlikely that a single clearinghouse for managing whitelisting is possible; it will more likely be unique to the source content owners and/or domains which implement DNS whitelists.

While this scenario may be unlikely, it may carry some benefits. First, parties performing troubleshooting would not have to determine whether or not DNS whitelisting was being used, as it always would be in use. In addition, if universally deployed, it is possible that the criteria for being added to or removed from a DNS whitelist could be standardized across the entire Internet. Nevertheless, even if uniform DNS whitelisting policies were not standardized, is also possible that a central registry of these policies could be developed and deployed in order to make it easier to discover them, a key part of achieving transparency regarding DNS whitelisting.

[EDITORIAL: Are there additional benefits or challenges to add here?]

## 5.2. Deploying DNS Whitelisting On An Ad Hoc Basis

This is the most likely deployment scenario for DNS whitelisting, as it seems today, is where some interested parties engage in DNS whitelisting but many or most others do not do so. What can make this scenario challenging from the standpoint of a DNS recursive resolver operator is determining which domains implement DNS whitelisting, particularly since a domain may not do so as they initially transition to IPv6, and may instead do so later. Thus, a DNS recursive resolver operator may initially believe that they can receive AAAA responses with IPv6 addresses as a domain adopts IPv6, but then notices via end user reports that they no longer receive AAAA responses due to that site adopting DNS whitelisting.

Thus, in contrast to universal deployment of DNS whitelisting, deployment on an ad hoc basis is likely to be significantly more challenging from an operational, monitoring, and troubleshooting standpoint. In this scenario, a DNS recursive resolver operator will have no way to systematically determine whether DNS whitelisting is or is not implemented for a domain, since the absence of AAAA records with IPv6 addresses may simply be indicative that the domain has not yet added IPv6 addressing for the domain, not that they have done so but have restricted query access via DNS whitelisting. As a result, discovering which domains implement DNS whitelisting, in order to differentiate them from those that do not, is likely to be challenging.

On the other hand, one benefit of DNS whitelisting being deployed on an ad hoc basis is that only the domains that are interested in doing so would have to upgrade their authoritative DNS servers in order to implement the ACLs necessary to perform DNS whitelisting.

[EDITORIAL: Additional benefits or challenges to add?]

## 6. What Problems Are DNS Whitelisting Implementers Trying To Solve?

As noted in Section 1, domains which implement DNS whitelisting are attempting to protect a few users of their domain, which happen to have impaired IPv6 access, from having a negative end user experience. While it is outside the scope of this document to explore the various reasons why a particular user may experience impaired IPv6 access, for the users which experience this it is a very real effect and would of course affect access to all or most IPv4 and IPv6 dual stack servers. This negative end user experience can range from someone slower than usual (as compared to native IPv4-based access), to extremely slow, to no access to the domain whatsoever.

Thus, parties which implement DNS whitelisting are attempting to provide a good experience to these end users. While one can debate whether DNS whitelisting is the optimal solution, it is quite clear that DNS whitelisting implementers are extremely interested in the performance of their services for end users as a primary motivation.

[EDITORIAL 1: More motivations to add?]

[EDITORIAL 2:Any good external references to consider adding?]

## 7. Implications of DNS Whitelisting

There are many potential implications of DNS whitelisting. In the sections below, the key potential implications are listed in some detail.

### 7.1. Architectural Implications

DNS whitelisting could be perceived as somewhat modifying the end-to-end model that prevails on the IPv4 Internet today. This approach moves additional access control information and policies into the middle of the network on the IPv6-addressed Internet, which did not exist before on the IPv4-addressed Internet. This could raise some risks noted in [RFC3724], which in explaining the history of the end-to-end principle [RFC1958] explains that one of the goals is to minimize the state, policies, and other functions needed in the middle of the network in order to enable end-to-end communications on the Internet.

It is also possible that DNS whitelisting could place at risk some of the benefits of the end-to-end principle, as listed in Section 4.1 of [RFC3724], such as protection of innovation. Further, while [RFC3234] details issues and concerns regarding so-called middleboxes, there may be parallels to DNS whitelisting, especially concerning modified DNS servers noted in Section 2.16 of [RFC3234], and more general concerns noted in Section 1.2 of [RFC3234] about the introduction of new failure modes, that configuration is no longer limited to two ends of a session, and that diagnosis of failures and misconfigurations is more complex.

In [Tussle in Cyberspace], the authors note concerns regarding the introduction of new control points, as well as "kludges" to the DNS, as risks to the goal of network transparency in the end-to-end model. Some parties concerned with the emerging use of DNS whitelisting have shared similar concerns, which may make [Tussle in Cyberspace] an interesting and relevant document. In addition, [Rethinking the design of the Internet] reviews similar issues that may be of

interest to readers of this document.

In order to explore and better understand these high-level architectural implications and concerns in more detail, the following sections explore more specific potential implications.

## 7.2. Public IPv6 Address Reachability Implications

The predominant experience of end user hosts and servers on the IPv4-addressed Internet today is that, very generally speaking, when a new server with a public IPv4 address is added, that it is then globally accessible by IPv4-addressed hosts. For the purposes of this document, that concept can be considered "pervasive reachability". It has so far been assumed that the same expectations of reachability would exist in the IPv6-addressed Internet. However, if DNS whitelisting is deployed, this will not be the case since only end user hosts using DNS recursive resolvers which have been added to the ACL of a given domain using DNS whitelisting would be able to reach new servers in that given domain via IPv6 addresses.

Thus, the expectation of any end user host being able to connect to any server (essentially both hosts, just at either end of the network), defined here as "pervasive reachability", will change to "restricted reachability" with IPv6.

[EDITORIAL: Additional implications?]

## 7.3. Operational Implications

This section explores some of the operationally related implications which may occur as a result of, related to, or necessary when engaging in the practice of DNS whitelisting.

### 7.3.1. De-Whitelisting May Occur

If it is possible for a DNS recursive resolver to be added to a whitelist, then it is also possible for that resolver to be removed from the whitelist, also known as de-whitelisting. Since de-whitelisting can occur, whether through a decision by the authoritative server operator or the domain owner, or even due to a technical error, an operator of a DNS recursive resolver will have new operational and monitoring requirements and/or needs as noted in Section 7.3.3, Section 7.3.4, Section 7.3.5, and Section 7.5.

### 7.3.2. Authoritative DNS Server Operational Implications

Operators of authoritative servers may need to maintain an ACL a server-wide basis affecting all domains, on a domain-by-domain basis,

as well as on a combination of the two. As a result, operational practices and software capabilities may need to be developed in order to support such functionality. In addition, processes may need to be put in place to protect against inadvertently adding or removing IP addresses, as well as systems and/or processes to respond to such incidents if and when they occur. For example, a system may be needed to record DNS whitelisting requests, report on their status along a workflow, add IP addresses when whitelisting has been approved, remove IP addresses when they have been de-whitelisted, log the personnel involved and timing of changes, schedule changes to occur in the future, and to roll back any inadvertent changes.

Such operators may also need implement new forms of monitoring in order to apply change control, as noted briefly in Section 7.3.4.

[EDITORIAL: Additional implications?]

### 7.3.3. DNS Recursive Resolver Server Operational Implications

Operators of DNS recursive resolvers, which may include ISPs, enterprises, universities, governments, individual end users, and many other parties, are likely to need to implement new forms of monitoring, as noted briefly in Section 7.3.4. But more critically, such operators may need to add people, processes, and systems in order to manage countless DNS whitelisting applications, for all domains that the end users of such servers are interested in now or in which they may be interested in the future. As such anticipation of interesting domains is likely infeasible, it is more likely that such operators may either choose to only apply to be whitelisted for a domain based upon one or more end user requests, or that they will attempt to do so for all domains.

When such operators apply for DNS whitelisting for all domains, that may mean doing so for all registered domains. Thus, some system would have to be developed to discover whether each domain has been whitelisted or not, which is touched on in Section 5 and may vary depending upon whether DNS whitelisting is universally deployed or is deployed on an ad hoc basis.

Furthermore, these operators will need to develop processes and systems to track the status of all DNS whitelisting applications, respond to requests for additional information related to these applications, determine when and if applications have been denied, manage appeals, and track any de-whitelisting actions. Given the incredible number of domains in existence, the ease with which a new domain can be added, and the continued strong growth in the numbers of new domains, readers should not underestimate the potential significance in personnel and expense that this could represent for



such operators. In addition, it is likely that systems and personnel may also be needed to handle new end user requests for domains for which to apply for DNS whitelisting, and/or inquiries into the status of a whitelisting application, reports of de-whitelisting incidents, general inquiries related to DNS whitelisting, and requests for DNS whitelisting-related troubleshooting by these end users.

[EDITORIAL: Additional implications?]

#### 7.3.4. Monitoring Implications

Once a DNS recursive resolver has been whitelisted for a particular domain, then the operator of that DNS recursive resolver may need to implement monitoring in order to detect the possible loss of whitelisting status in the future. This DNS recursive resolver operator could configure a monitor to check for a AAAA response in the whitelisted domain, as a check to validate continued status on the DNS whitelist. The monitor could then trigger an alert if at some point the AAAA responses were no longer received, so that operations personnel could begin troubleshooting, as outlined in Section 7.3.5.

Also, authoritative DNS server operators are likely to need to implement new forms of monitoring. In this case, they may desire to monitor for significant changes in the size of the whitelist within a certain period of time, which might be indicative of a technical error such as the entire ACL being removed. These operators may also wish to monitor their workflow process for reviewing and acting upon DNS whitelisting applications and appeals, potentially measuring and reporting on service level commitments regarding the time an application or appeal can remain at each step of the process, regardless of whether or not such information is shared with parties other than that authoritative DNS server operator.

These are but a few examples of the types of monitoring that may be called for as a result of DNS whitelisting, among what are likely many other types and variations.

[EDITORIAL: Additional implications?]

#### 7.3.5. Troubleshooting Implications

The implications of DNS whitelisted present many challenges, which have been detailed in Section 7. These challenges may negatively affect the end users' ability to troubleshoot, as well as that of DNS recursive resolver operators, ISPs, content providers, domain owners (where they may be different from the operator of the authoritative DNS server for their domain), and other third parties. This may make

the process of determining why a server is not reachable significantly more complex.

[SECTION INCOMPLETE - MIGHT LIKE TO ADD SOME EXAMPLES HERE]

[EDITORIAL: Additional implications?]

#### 7.3.6. Additional Implications If Deployed On An Ad Hoc Basis

[SECTION INCOMPLETE - IS THIS NEEDED? - PLACEHOLDER FOR NOW]

[EDITORIAL: Additional implications?]

#### 7.4. Homogeneity May Be Encouraged

A broad trend which has existed on the Internet appears to be a move towards increasing levels of heterogeneity. One manifestation of this is in an increasing number, variety, and customization of end user hosts, including home network, operating systems, client software, home network devices, and personal computing devices. This trend appears to have had a positive effect on the development and growth of the Internet. A key facet of this that has evolved is the ability of the end user to connect any technically compliant device or use any technically compatible software to connect to the Internet. Not only does this trend towards greater heterogeneity reduce the control which is exerted in the middle of the network, described in positive terms in [Tussle in Cyberspace], [Rethinking the design of the Internet], and [RFC3724], but it can also help to enable greater and more rapid innovation at the edges.

An unfortunate implication of the adoption of DNS whitelisting may be the encouragement of a reversal of this trend, which would be a move back towards greater levels of homogeneity. In this case, a domain owner which has implemented DNS whitelisting may prefer greater levels of control be exerted over end user hosts (which broadly includes all types of end user software and hardware) in order to attempt to enforce technical standards relating to establishing certain IPv6 capabilities or the enforcing the elimination of or restriction of certain end user hosts. While the domain operator is attempting to protect, maintain, and/or optimize the end user experience for their domain, the collective result of many domains implementing DNS whitelisting, or even a few important domains implementing DNS whitelisting, may be to encourage a return to more homogenous and/or controlled end user hosts. Unfortunately, this could have unintended side effects on and counter-productive implications for future innovation at the edges of the network.

## 7.5. Technology Policy Implications

A key technology policy implication concerns the policies relating to the process of reviewing an application for DNS whitelisting, and the decision-making process regarding whitelisting for a domain. Important questions may include whether these policies have been fully and transparently disclosed, are non-discriminatory, and are not anti-competitive. A related implication is whether and what the process for appeals is, when a domain decides not to add a DNS recursive resolver to the whitelist. Key questions here may include whether appeals are allowed, what the process is, what the expected turn around time is, and whether the appeal will be handled by an independent third party or other entity/group.

A further implications arises when de-whitelisting occurs. Questions that may naturally be raised in such a case include whether the criteria for de-whitelisting have been fully and transparently disclosed, are non-discriminatory, and are not anti-competitive. Additionally, the question of whether or not there was a cure period available prior to de-whitelisting, during which troubleshooting activities, complaint response work, and corrective actions may be attempted, and whether this cure period was a reasonable amount of time.

It is also conceivable that whitelisting and de-whitelisting decisions could be quite sensitive to concerned parties beyond the operator of the domain which has implemented DNS whitelisting and the operator of the DNS recursive resolver, including end users, application developers, content providers, advertisers, public policy groups, governments, and other entities, which may also seek to become involved in or express opinions concerning whitelisting and/or de-whitelisting decisions. Lastly, it is conceivable that any of these interested parties or other related stakeholders may seek redress outside of the process a domain has establishing for DNS whitelisting and de-whitelisting.

A final concern is that decisions relating to whitelisting and de-whitelisting may occur as an expression of other commercial, governmental, and/or cultural conflicts, given the new control point which has been established with DNS whitelisting. For example, in one imagined scenario, it may be conceivable that one government is unhappy with a news story or book published in a particular country, and that this government may retaliate against or protest this news story or book by requiring domains operating within that government's territory to de-whitelist commercial, governmental, or other entities involved in or related to (however tangentially) publishing the news story or book. By the same token, a news site operating in multiple territories may be unhappy with governmental policies in one

particular territory and may choose to express dissatisfaction in that territory by de-whitelisting commercial, governmental, or other entities in that territory. Thus, it seems possible that DNS whitelisting and de-whitelisting could become a vehicle for adjudicating other disputes, and that this may well have intended and unintended consequences for end users which are affected by such decisions and are unlikely to be able to express a strong voice in such decisions.

#### 7.6. IPv6 Adoption Implications

As noted in Section 3, the implications of DNS whitelisting may drive end users and/or networks to delay, postpone, or cancel adoption of IPv6, or to actively seek alternatives to it. Such alternatives may include the use of multi-layer network address translation (NAT) techniques like NAT444 [I-D.shirasaki-nat444], which these parties may decide to pursue on a long-term basis to avoid the perceived costs and aggravations related to DNS whitelisting. This could of course come at the very time that the Internet community is trying to get these very same parties interested in IPv6 and motivated to begin the transition to IPv6. As a result, parties concerned over the negative implications of DNS whitelisting have said they are very concerned of the negative effects that this practice could have on the adoption of IPv6 if it became widespread or was adopted by key parties in the Internet ecosystem.

[EDITORIAL: Additional implications?]

### 8. Solutions

#### 8.1. Implement DNS Whitelisting Universally

One obvious solution is to implement DNS whitelisted universally, and to do so using some sort of centralized registry of DNS whitelisting policies, contracts, processes, or other information. This potential solution seems unlikely at the current time.

[EDITORIAL: More to add?]

#### 8.2. Implement DNS Whitelisting On An Ad Hoc Basis

If DNS whitelisting was to be adopted more widely, it is likely to be adopted on this ad hoc, or domain-by-domain basis. Therefore, only those domains interested in DNS whitelisting would need to adopt the practice, though as noted herein discovering that they a given domain has done so may be problematic.

[EDITORIAL: More to add?]

### 8.3. Do Not Implement DNS Whitelisting

As an alternative to adopting DNS whitelisting, the Internet community can instead choose to take no action whatsoever, perpetuating the current predominant authoritative DNS operational model on the Internet, and leave it up to end users with IPv6-related impairments to discover and fix those impairments.

#### 8.3.1. Solving Current End User IPv6 Impairments

A further extension of not implementing DNS whitelisting, is to also endeavor to actually fix the underlying technical problems that have prompted the consideration of DNS whitelisting in the first place, as an alternative to trying to apply temporary workarounds to avoid the symptoms of underlying end user IPv6 impairments. A first step is obviously to identify which users have such impairments, which would appear to be possible, and then to communicate this information to end users. Such end user communication is likely to be most helpful if the end user is not only alerted to a potential problem but is given careful and detailed advice on how to resolve this on their own, or where they can seek help in doing so.

One challenge with this option is the potential difficulty of motivating members of the Internet community to work collectively towards this goal, sharing the labor, time, and costs related to such an effort. Of course, since just such a community effort is now underway for IPv6, it is possible that this would call for only a moderate amount of additional work.

[EDITORIAL: More to add?]

## 9. Security Considerations

There are no particular security considerations if DNS whitelisting is not adopted, as this is how the public Internet works today with A records.

However, if DNS whitelisting is adopted, organizations which apply DNS whitelisting policies in their authoritative servers should have procedures and systems which do not allow unauthorized parties to either remove whitelisted DNS resolvers from the whitelist or add non-whitelisted DNS resolvers to the whitelist. Should such unauthorized additions or removals from the whitelist can be quite damaging, and result in content providers and/or ISPs to incur substantial support costs resulting from end user and/or customer

contacts. As such, great care must be taken to control access to the whitelist for an authoritative server.

In addition, two other key security-related issues should be taken into consideration:

#### 9.1. DNSSEC Considerations

DNS security extensions defined in [RFC4033], [RFC4034], and [RFC4035] use cryptographic digital signatures to provide origin authentication and integrity assurance for DNS data. This is done by creating signatures for DNS data on a Security-Aware Authoritative Name Server that can be used by Security-Aware Resolvers to verify the answers. Since DNS whitelisting is implemented on an authoritative server, which provides different answers depending upon which resolver server has sent a query, the DNSSEC chain of trust is not altered. Therefore there are no DNSSEC implications per se, and thus no specific DNSSEC considerations to be listed.

#### 9.2. Authoritative DNS Response Consistency Considerations

[INCOMPLETE!!]

While Section 9.1 does not contain any specific DNSSEC considerations. However, it is certainly conceivable that security concerns may arise when end users or other parties notice that the responses sent from an authoritative DNS server appear to vary from one network or one DNS recursive resolver to another. This may give rise to concerns that, since the authoritative responses vary that there is some sort of security issue and/or some or none of the responses can be trusted.

#### 10. IANA Considerations

There are no IANA considerations in this document.

#### 11. Contributors

The following people made significant textual contributions to this document and/or played an important role in the development and evolution of this document:

John Brzozowski

Chris Griffiths

Tom Klieber

Yiu Lee

Rich Woundy

## 12. Acknowledgements

The authors and contributors also wish to acknowledge the assistance of the following individuals in helping us to develop and/or review this document:

## 13. References

### 13.1. Normative References

- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [RFC1958] Carpenter, B., "Architectural Principles of the Internet", RFC 1958, June 1996.
- [RFC2775] Carpenter, B., "Internet Transparency", RFC 2775, February 2000.
- [RFC2956] Kaat, M., "Overview of 1999 IAB Network Layer Workshop", RFC 2956, October 2000.
- [RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.
- [RFC3724] Kempf, J., Austein, R., and IAB, "The Rise of the Middle and the Future of End-to-End: Reflections on the Evolution of the Internet Architecture", RFC 3724, March 2004.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.
- [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

### 13.2. Informative References

- [I-D.shirasaki-nat444]  
Yamagata, I., Shirasaki, Y., Nakagawa, A., Yamaguchi, J.,  
and H. Ashida, "NAT444", draft-shirasaki-nat444-02 (work  
in progress), July 2010.
- [IETF 77 DNSOP WG Presentation]  
Gashinsky, I., "IPv6 & recursive resolvers: How do we make  
the transition less painful?", IETF 77 DNS Operations  
Working Group, March 2010,  
<<http://www.ietf.org/proceedings/77/slides/dnsop-7.pdf>>.
- [Network World Article on DNS Whitelisting]  
Marsan, C., "Google, Microsoft, Netflix in talks to create  
shared list of IPv6 users", Network World , March 2010, <<http://www.networkworld.com/news/2010/032610-dns-ipv6-whitelist.html>>.
- [Network World Article on IETF 77 DNSOP WG Presentation]  
Marsan, C., "Yahoo proposes 'really ugly hack' to DNS",  
Network World , March 2010, <<http://www.networkworld.com/news/2010/032610-yahoo-dns.html>>.
- [Rethinking the design of the Internet]  
Blumenthal, M. and D. Clark, "Rethinking the design of the  
Internet: The end to end arguments vs. the brave new  
world", ACM Transactions on Internet Technology Volume 1,  
Number 1, Pages 70-109, August 2001, <[http://dspace.mit.edu/bitstream/handle/1721.1/1519/TPRC\\_Clark\\_Blumenthal.pdf](http://dspace.mit.edu/bitstream/handle/1721.1/1519/TPRC_Clark_Blumenthal.pdf)>.
- [Tussle in Cyberspace]  
Braden, R., Clark, D., Sollins, K., and J. Wroclawski,  
"Tussle in Cyberspace: Defining Tomorrow's Internet",  
Proceedings of ACM Sigcomm 2002, August 2002, <<http://groups.csail.mit.edu/ana/Publications/PubPDFs/Tussle2002.pdf>>.

### Appendix A. Document Change Log

- [RFC Editor: This section is to be removed before publication]
- 00: First version published
- 01: Updated the title of the document, to avoid confusion (based on  
feedback)



## Appendix B. Open Issues

[RFC Editor: This section is to be removed before publication]

1. Incorporate any feedback received at IETF 79
2. Incorporate feedback from Erik Kline, received 10/1/2010
3. Incorporate feedback from Brian Carpenter, received 10/19/2010
4. Bring on new contributors: Hannes Tschofenig and Danny McPherson has so far offered to contribute.
5. Close out any EDITORIAL notes
6. Add any good references throughout the document
7. Add reviewers to the acknowledgements section
8. Ensure references are in the proper section (normative/informative)
9. Include a number of references from RFC3724?
10. Call DNS WL something else or add note to the effect that this is unrelated to DNS WL used for email - such as [www.dnswl.org](http://www.dnswl.org)

## Author's Address

Jason Livingood  
Comcast Cable Communications  
One Comcast Center  
1701 John F. Kennedy Boulevard  
Philadelphia, PA 19103  
US

Email: [jason\\_livingood@cable.comcast.com](mailto:jason_livingood@cable.comcast.com)  
URI: <http://www.comcast.com>



Internet  
Internet-Draft  
Intended status: Standards Track  
Expires: March 26, 2011

N. Shen  
C. Pignataro  
R. Asati  
E. Chen  
Cisco Systems, Inc.  
A. Atlas  
BT  
September 22, 2010

Traceroute and Ping Message Extension  
draft-shen-traceroute-ping-ext-00

Abstract

This document specifies an extension to traceroute and ping messages that allows the probe packets to be authenticated by the intermediate nodes and the destination node. This extension can also include requests for node specific information that the probe sender is interested to receive from one or more nodes via the traceroute and ping replies. This extension supports UDP, TCP and ICMP types of traceroute and ping probes.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 26, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Specification of Requirements . . . . .	3
2. Introduction . . . . .	3
3. Motivation . . . . .	3
4. Probe Message Extension . . . . .	4
4.1. Probe Structure . . . . .	4
4.1.1. Probe Common Header . . . . .	4
4.1.2. Probe TLV . . . . .	5
4.1.2.1. Probe Authentication TLV . . . . .	5
4.1.2.2. Probe Information-Request TLV . . . . .	7
4.2. Probe Extension Offset Field . . . . .	7
4.2.1. UDP Messages . . . . .	8
4.2.2. TCP Messages . . . . .	8
4.2.3. ICMP Messages . . . . .	8
4.2.4. Implementation Discussion . . . . .	8
5. Implementation and Operation Considerations . . . . .	9
5.1. Traceroute and Ping Probe Sender . . . . .	9
5.2. Traceroute and Ping Probe Receiver . . . . .	9
6. Security Considerations . . . . .	10
7. IANA Considerations . . . . .	10
8. Acknowledgements . . . . .	10
9. References . . . . .	11
9.1. Normative References . . . . .	11
9.2. Informative References . . . . .	11
Authors' Addresses . . . . .	12

## 1. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 2. Introduction

Traceroute and Ping are tools widely used in the diagnosis of network problems. This document proposes the mechanism in which the source probe packets can be authenticated by the intermediate and the destination nodes. It also specifies the mechanism for specific information in the traceoute and ping replies the source is interested to get. This mechanism gives operators more secure ways of performing network management and troubleshooting tasks. It applies to both IPv4 and IPv6 networks.

This document applies to most type of traceroute and ping probe packets including UDP [RFC0768], TCP [RFC0793] and ICMP/ICMPv6 [RFC0792] [RFC4443] types. The ICMP/ICMPv6 reply messages responding to those probes are not part of this specification.

This document defines an extension for traceroute and ping probe messages to optionally include authentication signature. The intermediate and destination nodes can authenticate the sender of the traceroute or ping packet before providing the requested information in the ICMP response. This document also includes an Information-Request TLV for the traceroute/ping extension. This TLV specifies the types of information the sender expects to be included in the traceroute/ping response (i.e., in the ICMP message elicited by the traceroute/ping packet and generated by the intermediate or destination node or nodes).

This specification is evolved from the UDP Traceroute Extension document [I-D.shen-udp-traceroute-ext].

## 3. Motivation

Although one may employ a rudimentary control mechanism to limit the trusted senders by defining on every router the access control lists specifying source addresses of the traceroute and ping message, such mechanism is deemed configuration intensive, static, and error-prone. Moreover, such mechanism would be susceptible to address spoofing. Additionally, such mechanism does not provide the sender with dynamic control of the different kind of extensions to be requested.

The ICMP reply messages has been extended to support multi-part message inside ICMP [RFC4884]. Some of the applications [RFC5837] [RFC4950] [I-D.shen-icmp-routing-inst] are designed mainly for internal network troubleshooting by network operators. Network providers may want to limit those applications only to trusted senders of traceroute/ping probes due to security or policy reasons by using this mechanism described in this document.

#### 4. Probe Message Extension

This proposed extension is to define a probe data structure which resides within UDP/TCP/ICMP data field; and to reserve the lowest 4 bits inside a 16-bit field within UDP/TCP/ICMP headers to indicate the extension structure offset location.

In most of the traceroute implementation, there is some private data in probe messages used by traceroute applications. With this "extension offset" defined, the applications can continue to use those private data while supporting this probe extension in a deterministic way. This extension applies to both traceroute and ping applications.

##### 4.1. Probe Structure

The probe structure starts in UDP/TCP/ICMP data field location from 0 to 56 octets specified in the "extension offset", see Section 4.2, in 32-bit boundary. It MUST have exactly one probe common header followed by zero or more probe TLVs.

##### 4.1.1. Probe Common Header

The Common Header is a 8 octets structure has the following format:

0								1								2								3							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version								Length								Checksum															
								Magic-Number (0x54726163)																							

The fields of the Common Header are defined as follows:

Version: 4 bits. It is defined as 1 in this document.

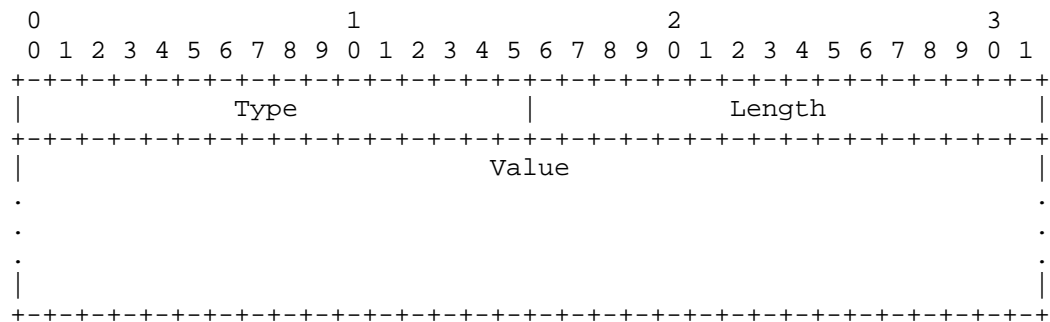
Length: 12 bits. The total length of the probe data structure specifying number of 32-bit words (includes the common header and all the TLVs).

Checksum: 16 bits. The one's complement of the one's complement sum of the probe data structure, with the checksum field replaced by zero for the purpose of computing the checksum.

Magic Number: 32 bits. It is defined as Hex value of 0x54726163 in this document. This is used mainly for structure identification of this extension version.

#### 4.1.2. Probe TLV

Probe TLVs (Type-Length-Value tuples) have the following format:



Type: 16 bits.

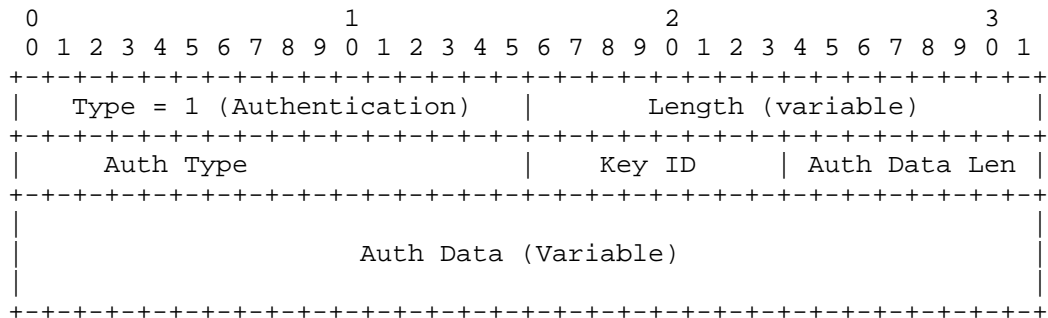
Length: 16 bits. Length of the Value field in octets.

Value: Depends on the Type. It is zero padded to align to a 4-octet boundary.

This document defines two TLVs below.

##### 4.1.2.1. Probe Authentication TLV

This TLV carries the HMAC authentication related information. It verifies both the data integrity and the authenticity of the entire message. This TLV has the following format:



Auth Type: 16 bits. The following values are proposed:

- \* Type=0 signifies no authentication.
- \* Type=1 signifies simple password based authentication.
- \* Type=2 signifies Cryptographic authentication.

Please note that the above type values are in line with IANA allocated values for other protocols (e.g., OSPF).

Key ID: 8 bits. This allows multiple secret keys to be active simultaneously. Using Key IDs makes the key rollover convenient. Each secret key must be associated with the hash algorithm. This may be done through provisioning on each node.

Auth Data Len: 8 bits. This specifies the length of the authentication data (and allows for the support of current and future authentication schemes).

Auth Data: Variable length. This field carries the result (e.g., HMAC code) of the HMAC algorithm applied over the entire traceroute/ping IP/IPv6 packet. When the Auth data is calculated, the shared key is stored in this field, and the checksum fields in the IP header, UDP/TCP/ICMP header and probe common header are set to zero. The result of the algorithm is placed in the Auth Key field. The following lists algorithms that could be commonly supported:

- \* HMAC-MD5
- \* HMAC-SHA1
- \* HMAC-SHA2 variants (e.g., 224, 256, 384, 512, etc.)



At least HMAC-MD5 and HMAC-SHA1 algorithms should be supported on all the nodes compliant with this specification.

#### 4.1.2.2. Probe Information-Request TLV

The Information-Request TLV has the following format:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Type = 2 (Info-Req)										Length = 4																													
Info Request																																							

Info-Req: 32 bits. This bitflag field lists the request items the probe sender is interested. The bit number ranges from the right most bit to the left most bit. Currently defined as the following:

Bit	Number	Information Item
0		MPLS label related attributes
1		Interface related attributes
2		IP/IPv6 address related attributes
3		Routing Instance related attributes
4		Nexthop(s) related attributes
5		Device role related attributes

## 4.2. Probe Extension Offset Field

This probe "extension offset" field is defined as the lowest nibble within a 16-bit field, and it specifies the position at which the probe extension data structure begins. The value represents 32-bit words ranges from 0x0 to 0xF, with value 0xF as reserved. Thus the position of the probe data structure can start from 0 to 56 octets inside TCP, UDP or ICMP data field. The probe "extension offset" field value 0xF indicates there is no probe data structure inside the message data field.

The probe "extension offset" field is defined as the following:

```

      0                               1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+
|                               |Ext-Off|
+---+---+---+---+---+---+---+---+

```

Ext-Off: 4 bits. The value (Ext-Off) represents the probe data structure start position in 32-bit words. The Ext-Off value 0xF is reserved.

The rest of the 12 bits out of this 16-bit field is not changed by this proposal. For application usage detail in terms of different traceroute/ping probe types, see Section 4.2.4.

#### 4.2.1. UDP Messages

In the UDP traceroute/ping probe case, this 16-bit field is the UDP source port field in UDP header [RFC0768]. The "Ext-Off" specifies the probe extension structure start location inside UDP data field.

#### 4.2.2. TCP Messages

In the TCP traceroute/ping probe case, this 16-bit field is the TCP source port field in TCP header [RFC0793]. The "Ext-Off" specifies the probe extension structure start location inside TCP data field.

#### 4.2.3. ICMP Messages

In the ICMP traceroute/ping probe case, this 16-bit field is the "Identifier" field of ICMP type 8 structure [RFC0792], The "Ext-Off" specifies the probe extension structure start location inside ICMP data field of the type 8(ICMP echo request) message.

#### 4.2.4. Implementation Discussion

In the majority of today's traceroute implementations, the application process identifier (process-ID) is used as the UDP source port for UDP type of traceroute probe; in TCP implementation, the process-ID is used as the TCP source port for traceroute probe; and the process-ID is also used as the ICMP Identifier of ICMP type 8 message. With this extension, an implementation can use the highest 12-bits of the source port field for UDP/TCP header and ID field for ICMP type 8 message to encode this process information since the lowest 4-bits are now reserved for the probe "extension offset".

Ping implementation is similar to traceroute, it either uses the process-ID or an internally generated number inside the ICMP echo request ID field and in UDP/TCP source port field. An implementation

now can use the highest 12-bits of the field and leave the lowest 4-bits for the probe extension.

## 5. Implementation and Operation Considerations

There is no change in this extension for the normal traceroute/ping implementation and operation except for reserving the lowest 4 bits in the UDP/TCP source port field and ICMP Id field of type 8 message. The implementations for the sender can use the same semantics with this 16-bit field; and it makes no difference to the receivers if they don't support this extension.

### 5.1. Traceroute and Ping Probe Sender

The sender supports this extension MAY include the Probe structure in it's traceroute/ping probe to specify the request types and authentication key. The sender SHOULD set the "extension offset" value to 0xF if there is no Probe structure present inside the probe. The sender MAY request one or multiple types of information defined in the probe "Info-Req" TLV.

### 5.2. Traceroute and Ping Probe Receiver

When the traceroute/ping probe receiver, the intermediate and destination node, processes the incoming probe, it MAY check the Probe structure to verify if the sender is from an authenticated host and to see what types of information it requested. This check is only needed when the receiver tries to authenticate the probe sender, or when the receiver is forming the ICMP and ICMPv6s that support multi-part messages and it has certain internal information that can be included in the ICMP packets.

If the probe "extension offset" value is not 0xF, the probe structure may be present. The receiver MUST verify the integrity of the data structure by examining the "version" field, the Magic-Number value, and the length of the probe structure. It MUST perform the checksum to verify the probe data structure. If the authentication TLV is present and the local policy requires it to perform the verification, the receiver MUST use it's locally stored shared key to validate the checksum in the TLV. Multiple Authentication Keys can be used which can be useful in the case the probes are from trusted peer networks.

If the probe "Info-Req" TLV is included, the receiver SHOULD fetch the related information when forming the ICMP packets, but MUST NOT include information that has the corresponding bitflag cleared.

Even if the authentication fails, the receiver MAY still send the

regular ICMP echo reply back to the sender, without the requested or internal information, as if this probe extension is not supported.

## 6. Security Considerations

This extension enhances the security of traceroute and ping operation in a backwards-compatible fashion. The mechanism allows the receiver to verify the sender of the traceroute/ping packet such that certain sensitive interface and network related information can be supplied in the internal network or across trusted networks.

The use of Cryptographic authentication (i.e., an Auth Type value of 2) allows for a strong authentication mechanism since the keys cannot be discerned by intercepting the packets. The proposed Keyed authentication does not prevent replay attacks. However, in the case of replay attacks, since the packet source IP/IPv6 address of the traceroute/ping probe can not be changed, there is no easy way for the attacker to retrieve the ICMP messages.

A router needs to protect against purposefully-bogus Traceroute packets with extensions that fail the authentication, as a high rate of messages can require significant processing time. [RFC1812] specifies how rate-limiting is applied to the generation of ICMP messages, and this rate-limiting deters the threat when applied before checking the Authentication. Additionally, when using Cryptographic authentication, the HMAC includes the source IP address, which means the HMAC will not validate if the traceroute/ping packet is sent over a NAT.

## 7. IANA Considerations

The Probe Extension contains probe TLVs. IANA should establish a registry of Probe Extension Types. This document defines Type 1 and Type 2 for authentication and information-request. Types 3-0xF6 are allocated through Expert Review [RFC5226]. Types 0xF7 to 0xFF are reserved for private use.

IANA should also establish a registry for Probe Info-Request Bits. This document defines bits 0 - 5 in Section 4.1.2.2. Bits 6-29 are allocated through Expert Review. Bits 30 - 31 are reserved for private use.

## 8. Acknowledgements

Many thanks to Dan Wing and Tony Li, for their insightful comments

and valuable suggestions regarding this document.

## 9. References

### 9.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

### 9.2. Informative References

- [I-D.shen-icmp-routing-inst]  
Shen, N. and E. Chen, "ICMP Extensions for Routing Instances", draft-shen-icmp-routing-inst-00 (work in progress), November 2006.
- [I-D.shen-udp-traceroute-ext]  
Shen, N., Pignataro, C., Asati, R., and E. Chen, "UDP Traceroute Message Extension", draft-shen-udp-traceroute-ext-01 (work in progress), June 2008.
- [RFC1812] Baker, F., "Requirements for IP Version 4 Routers", RFC 1812, June 1995.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.
- [RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, April 2007.
- [RFC4950] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "ICMP Extensions for Multiprotocol Label Switching", RFC 4950, August 2007.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an

IANA Considerations Section in RFCs", BCP 26, RFC 5226,  
May 2008.

[RFC5837] Atlas, A., Bonica, R., Pignataro, C., Shen, N., and JR.  
Rivers, "Extending ICMP for Interface and Next-Hop  
Identification", RFC 5837, April 2010.

#### Authors' Addresses

Naiming Shen  
Cisco Systems, Inc.  
225 West Tasman Drive  
San Jose, CA 95134  
USA

Email: [naiming@cisco.com](mailto:naiming@cisco.com)

Carlos Pignataro  
Cisco Systems, Inc.  
7200 Kit Creek Road  
Research Triangle Park, NC 27709  
USA

Email: [cpignata@cisco.com](mailto:cpignata@cisco.com)

Rajiv Asati  
Cisco Systems, Inc.  
7025 Kit Creek Road  
Research Triangle Park, NC 27709  
USA

Email: [rajiva@cisco.com](mailto:rajiva@cisco.com)

Enke Chen  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134  
USA

Email: [enkechen@cisco.com](mailto:enkechen@cisco.com)

Alia K. Atlas  
BT

Email: [alia.atlas@bt.com](mailto:alia.atlas@bt.com)





Softwire  
Internet-Draft  
Intended status: Standards Track  
Expires: January 30, 2011

O. Vautrin  
Juniper Networks  
July 29, 2010

IPv4 Rapid Deployment on IPv6 Infrastructures (4rd)  
draft-vautrin-softwire-4rd-00

Abstract

This document specifies an automatic tunneling mechanism tailored to advance deployment of IPv4 to end users via an IPv6 network infrastructure. This document aims at giving an alternative to family translation to operate an Ipv6-only network.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 30, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

## Table of Contents

1. Introduction . . . . .	3
2. Requirements Language . . . . .	4
3. Terminology . . . . .	4
4. 4rd Model and operation . . . . .	5
4.1. Traffic from CE to IPv4 Internet [CE Behavior] . . . . .	5
4.2. Traffic from CE to IPv4 Internet [BR Behavior] . . . . .	6
4.3. Traffic from IPv4 Internet to CE [CE Behavior] . . . . .	6
4.4. Traffic from IPv4 Internet to CE [BR Behavior] . . . . .	6
5. IPv6-only Deployment considerations . . . . .	6
6. Acknowledgements . . . . .	7
7. IANA Considerations . . . . .	7
8. Security Considerations . . . . .	7
9. Normative References . . . . .	7
Author's Address . . . . .	8

## 1. Introduction

4rd specifies a protocol mechanism to deploy IPv4 to sites or Host via an IPv6 network. It builds on [I-D.ietf-softwire-ipv6-6rd] and [I-D.ietf-softwire-dual-stack-lite]. 4rd could be seen either as the opposite of [I-D.ietf-softwire-ipv6-6rd] or as [I-D.ietf-softwire-dual-stack-lite] without NAT (or leaving NAT as optional).

IPv6-only network are not common. But Ipv6-only networks is the end goal in the Ipv4 to IPv6 transition. Thus it is worthwhile to define viable mechanism to ease the use of Ipv6-only network. The alternatives to 4rd are defined in [I-D.ietf-behave-v6v4-framework] and such mechanisms have well known limitation most of them described in [RFC4966].

The 4rd mechanism relies upon a tunneling of IPv4 inside IPv6 to a well known IPv6 address to allow automatic IPv4 operation in an IPv6-only Network. The mechanism can be stateless or stateful depending on the selection of the IPv6 address. If the Ipv6 address is using the IPv4-Embedded IPv6 Address Format described in [draft-ietf-behave-address-format] then the 4rd operation will be stateless. If the algorithmic mapping is not used, 4rd will fall back to a Standard DS-Lite operation. 4rd views the IPv6 network as a link layer for IPv4 and supports an automatic tunneling abstraction similar to the Non-Broadcast Multiple Access (NBMA) [RFC2491] model.

A 4rd domain consists of 4rd Customer Edges (CE) and one or more 4rd Border Relays (BRs). IPv4 packets encapsulated by 4rd follow the IPv6 routing topology within the network among CEs and BRs. 4rd BRs are traversed only for IPv4 packets that are destined to or are arriving from outside the 4rd domain. The CE can be either a host (which would need to have a 4rd client capability) or a router (On the LAN side of the router, IPv4 is implemented as it would be for any native IP service delivered by the network).

4rd relies on IPv6 and is designed to deliver production-quality IPv4 alongside IPv6 with as little change to IPv6 networking and operations as possible. 4rd can be deployed and thus remove the need for a Dual stack Network completely helping the transition to a full IPv6 internet in the future.

4rd used with a short IPv4 DHCP lease time or in conjunction with NAT44 (DS-Lite) can also be seen as an IPv4-depletion mitigation solution.

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 3. Terminology

**4rd\_IPv4\_prefix** - An IPv4 prefix selected for use by a 4rd domain. There is exactly one 4rd IPv4-prefix for a given 4rd domain. A network may deploy 4rd with a single 4rd domain or multiple 4rd domains.

**4rd Customer Edge** - A 4rd CE is a device functioning as a Customer Edge in a 4rd deployment. A 4rd CE may also be referred simply as a "CE" within the context of 4rd.

**4rd domain** - A set of 4rd CEs and BRs connected to the same virtual 4rd link.

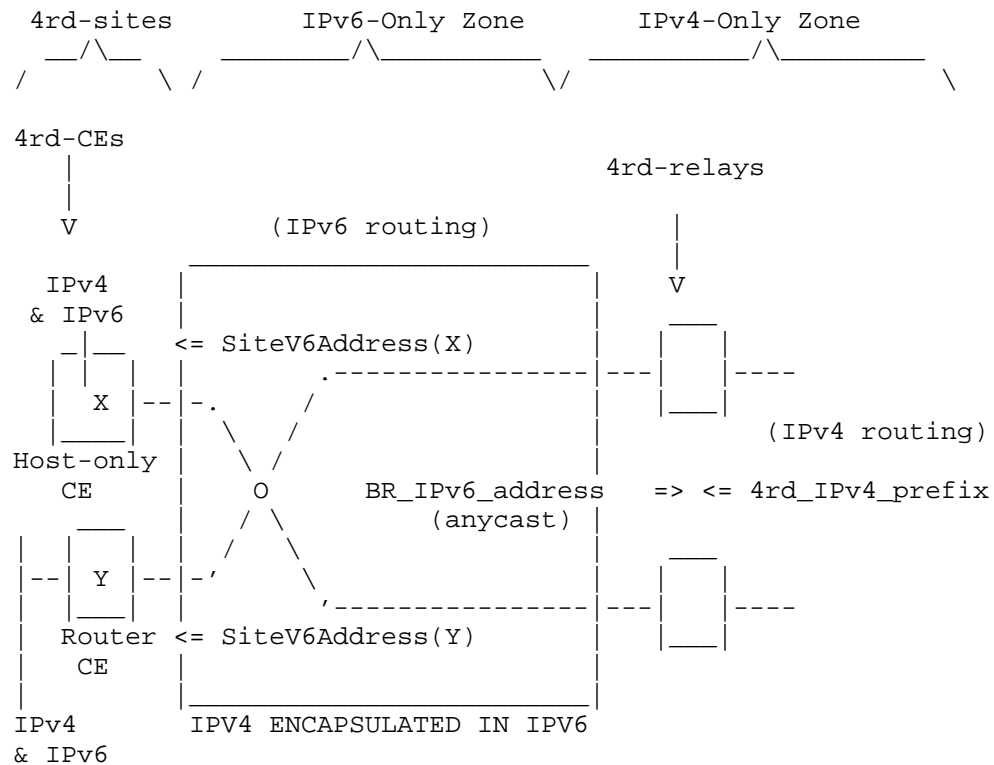
**4rd Border Relay (BR)** - A 4rd-enabled router managed at the edge of a 4rd domain. A border relay router has at least one of each of the following: an IPv6-enabled interface, a 4rd virtual interface acting as an endpoint for the 4rd IPv4 in IPv6 tunnel, and an IPv4 interface connected to the native IPv4 network. A 4rd BR may also be referred to simply as a "BR" within the context of 4rd.

**BR\_IPv6\_address** - The IPv6 address of the 4rd Border Relay for a given 4rd domain. This IPv6 address is used by the CE to send packets to a BR in order to reach IPv4 destinations outside of the 4rd domain.

**CE\_IPv6\_address** - The IPv6 address given to the CE through normal means (i.e., configured via DHCP, or otherwise). With proper DHCP and Network design planning, this address can match the CE\_IPv4\_address that the CE will receive and thus use an IPv4-Embedded IPv6 Address Format described in [draft-ietf-behave-address-format]).

**CE\_IPv4\_address** - The IPv4 address given to the CE through the IPv6 tunnel (i.e., configured via DHCP, or otherwise). This means the CE can only get its CE\_IPv4\_address when it already has an CE\_IPv6\_address. This address may be global or private [RFC1918]. This address is used to send and receive IPv4 packets.

## 4. 4rd Model and operation



THE 4RD MODEL

Figure 1

## 4.1. Traffic from CE to IPv4 Internet [CE Behavior]

the CE encapsulate the IPv4 packet into an IPv6 tunnel (aka Software). The IPv4 source packet can be either private or public. It can be learned through the IPv6 tunnel or by other means. The IPv6 source address can be either an IPv4-Embedded IPv6 Address or not. The choice to use IPv4-Embedded IPv6 Address or not will have an impact on the BR as this will switch between the stateless mode or the stateful mode.

#### 4.2. Traffic from CE to IPv4 Internet [BR Behavior]

If the IPv6 packet source address is using an IPv4-Embedded IPv6 Address, then in this direction the BR just decapsulate the IPv4 packets from the IPv6 tunnel and forward it to the IPv4 Internet. This is what we call the Stateless 4rd mode. If the CE\_IPv6\_address is *\*not\** using an IPv4-Embedded IPv6 Address, then the BR need to keep track of the relationship of this IP session and the Ipv6 tunnel. The IPv6 address becomes the ID of the session. This is what we call the Stateful 4rd mode.

The BR is either doing NAT44 with the IPv6 address as the host identifier if the CE\_IPv4\_address is a private address or the BR is creating a mapping table between the software ID and the CE\_IPv4\_address if this last one is public and should not be modified. Note that 1:N NATP can be used in parallel either on the same device or on another one. This mechanism is then similar to DS-Lite.

#### 4.3. Traffic from IPv4 Internet to CE [CE Behavior]

The CE decapsulate the IPv4 packets from the IPv6 packets.

#### 4.4. Traffic from IPv4 Internet to CE [BR Behavior]

If a session or a mapping information already exist in the system that matches the IPv4 packets, the IPv6 packets will be created with the information based on this session information. The session can exist because of traffic that originated from the Ipv6 side or because some Port or address forwarding have been configured on the BR. If no sessions exist, the stateless mechanism will be used and the IPv6 packets will be created using the IPv4 address as defined by the IPv4 Mapped address mapping.

### 5. IPv6-only Deployment considerations

- Scenario 1: Service Provider with IPv6-only access would like to give an IPv4 address to end subscribers.

4rd used with a short IPv4 DHCP lease time or in conjunction with NAT44 (DS-Lite) can also be seen as an IPv4-depletion mitigation solution. With more and more internet content accessible through IPv6, An Ipv4 address could be needed in the future just to access some legacy content. This means an Ipv4 address could be needed only temporarily. This means temporary allocation of Ipv4 addresses with short lease time can be a useful IPv4-depletion mitigation solution.

- Scenario 2: An IPv6-only Enterprise would like to give IPv4 connectivity.

In this case, operating systems would have to support 4rd the same way current operating systems support 6to4, Teredo or ISATAP. An alternative would be to deploy island of IPv4 with 4rd Clients running on routers.

- Scenario 3: An IPv6-only Enterprise would like to restore their servers connectivity from IPv4 Internet. In this case, the 4rd client will be started either on the server itself or on the 1st hop router.

## 6. Acknowledgements

None

## 7. IANA Considerations

None

## 8. Security Considerations

To be defined.

## 9. Normative References

[I-D.ietf-behave-v6v4-framework]

Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", draft-ietf-behave-v6v4-framework-09 (work in progress), May 2010.

[I-D.ietf-softwire-dual-stack-lite]

Durand, A., Droms, R., Haberman, B., Woodyatt, J., Lee, Y., and R. Bush, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-05 (work in progress), July 2010.

[I-D.ietf-softwire-ipv6-6rd]

Townsley, M. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", draft-ietf-softwire-ipv6-6rd-10 (work in progress), May 2010.

- [RFC1990] Sklower, K., Lloyd, B., McGregor, G., Carr, D., and T. Coradetti, "The PPP Multilink Protocol (MP)", RFC 1990, August 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

Author's Address

Olivier Vautrin  
Juniper Networks  
1194 N Mathilda Avenue  
Sunnyvale, CA 94089  
USA

Email: Olivier@juniper.net



